

VISIT...

LANZAROTE
Caliente.COM

519
M-75
57823

И. Н. МОЛЧАНОВ

МАШИННЫЕ
МЕТОДЫ РЕШЕНИЯ
ПРИКААДНЫХ ЗАДАЧ.
АЛГЕБРА,
ПРИБЛИЖЕНИЕ
ФУНКЦИЙ

АКАДЕМИЯ НАУК УКРАИНСКОЙ ССР
ИНСТИТУТ КИБЕРНЕТИКИ имени В. М. ГЛУШКОВА

И. Н. МОЛЧАНОВ

МАШИННЫЕ
МЕТОДЫ РЕШЕНИЯ
ПРИКЛАДНЫХ ЗАДАЧ
АЛГЕБРА,
ПРИБЛИЖЕНИЕ
ФУНКЦИЙ

КИЕВ НАУКОВА ДУМКА 1987

Машинные методы решения прикладных задач. Алгебра, приближение функций / Молчанов И. Н. — Киев : Наук. думка, 1987. — с. 288.

Монография посвящена численным методам решения задач линейной алгебры, нелинейных уравнений, приближению функций.

Наряду с алгоритмами решения задач и их теоретическим обоснованием рассмотрены проблемы машинной реализации численных методов решения прикладных задач на ЭВМ и встречающиеся при этом трудности.

На примерах показано, что формальная реализация на ЭВМ теоретически обоснованных с точки зрения классической математики алгоритмов в ряде случаев может давать машинное решение задачи, отличающееся от решения прикладной задачи.

Рассчитана на специалистов-прикладников, занимающихся численным решением научно-технических задач на ЭВМ, а также аспирантов и студентов старших курсов как математических, так и технических факультетов вузов.

Ил. 20. Табл. 17. Библиогр.: с. 278—285 (193 назв.).

Ответственный редактор *А. Н. Химич*

Рецензенты *А. А. Глуценко, Б. Н. Пшеничный*

Редакция физико-математической литературы

Внедрение вычислительной техники во все области деятельности человека является характерной чертой развивающейся научно-технической революции. Возможности современных электронных вычислительных машин (ЭВМ), накопленный опыт решения с их помощью сложных задач, а также достаточно высокий уровень развития математики позволяют сделать численный эксперимент одним из средств научного и инженерного исследования.

Под численным экспериментом понимают исследование свойств объекта или явления с помощью решения на ЭВМ уравнений, представляющих собой математическую модель объекта или явления. Задавая различными способами исходные данные и решая на основе этих данных уравнения, можно понять роль и значение различных факторов, определяющих изучаемые объекты или явления.

Организация и проведение численного эксперимента выдвигают требования достоверности получаемых на ЭВМ решений. Проблема достоверности содержит два естественных аспекта: достоверность математической модели, описывающей прикладную задачу, и достоверность машинного решения уравнений математической модели изучаемого объекта или явления. Ставится задача оценки по заданной априорной и получаемой в ходе решения апостериорной информации близости машинного к математическому и математического к физическому решениям задач.

Для большинства рассматриваемых классов задач на элементарных примерах показано, как на основе физических закономерностей формируются физические и математические модели задач, вводятся необходимые обозначения и определения, а также используемый в дальнейшем математический аппарат, и как формальное применение математического аппарата, не учитывающего специфики машинной реализации, может привести к ошибочным результатам.

Цель книги — показать построение численных методов решения классических уравнений, возникающих в каждой из рассматриваемых задач, дать основные идеи и понятия, а также некоторые навыки в исследованиях, которые необходимы для решения задач, возникающих на практике. В монографии описаны построения и некоторые исследования алгоритмов, сформулированы условия применения численных методов, дана их краткая характеристика. Но говорить о методах решения, не учитывая условий, при которых построена математическая модель (погрешность задания исходных данных, существование и единственность решения задач и т. д.), по существу невозможно. В то же время не всякое машинное решение задачи есть решение уравнений математической или физической модели, поэтому в книге рассматриваются некоторые вопросы анализа результатов счета, полученных на машинах, даются некоторые рекомендации по проверке точности решения. Значительная часть методов для лучшего понимания проиллюстрирована несложными примерами.

Каждая глава монографии, как правило, содержит постановку задачи; некоторые сведения справочного характера из других разделов математики, используемые при изложении материала; основные понятия и идеи в каждом классе задач; численные методы решения; аппарат исследования численных методов, который может быть использован при решении нестандартных (неклассических) задач (применение аппарата иллюстрируется стандартными задачами); средства контроля достоверности получаемых решений; заключительные замечания, оценивающие современное состояние и перспективы методов решения каждого класса задач.

Предлагаемая монография является отражением методики вычислений, сложившейся в отделе программирования и решения задач Института кибернетики имени В. М. Глушкова АН УССР.

Автор вместе с сотрудниками в течение ряда лет занимался решением задач практического значения, а также работами по созданию математического обеспечения ЭВМ.

Автор признателен академику А. А. Дородницину за ценные советы, определившие принцип построения монографии.

Большую помощь в работе над монографией оказали Л. Д. Николенко, Е. Ф. Галба, Г. И. Визнюк, А. Н. Химич, В. С. Зубатенко, И. В. Решетуа, Н. И. Степанец, О. П. Бурдаков. Ценные советы и замечания были высказаны Х. Д. Икрамовым, А. И. Степанцом.

Работа по оформлению рукописи и рисунков выполнена Т. А. Лиходзневской, Л. И. Милевской, И. П. Винокуровой, Т. А. Герасимовой, Н. В. Лапс.

Всем, способствовавшим появлению этой книги, автор выражает глубокую благодарность.

ГЛАВА I

ПРЯМЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

1.1. ПОСТАНОВКА ЗАДАЧИ, ОПРЕДЕЛЕНИЯ, ОБЩИЕ СВОЙСТВА

1. Задача о распределении токов в замкнутой электрической цепи. Многие проблемы в различных областях науки и техники приводят к решению одной и той же математической задачи — решению системы линейных алгебраических уравнений. Рассмотрим на частном примере, как возникают системы линейных алгебраических уравнений.

Пусть требуется определить распределение токов в электрической схеме (рис. 1) без амперметра. Сопротивления участков цепи R_1 , R_2 , R_3 и электродвижущая сила источников тока (ЭДС) E_1 и E_2 известны. Положительные направления токов показаны на рисунке стрелками. Если бы был амперметр, то, измерив силу тока в отдельных участках цепи, можно было бы получить физическое решение задачи. А так как по условиям задачи его нет, то для определения силы тока используем известные физические закономерности и математические средства. Напомним известные законы Кирхгофа. Первый из них относится к узлам цепи. Узлом называется точка, в которой сходятся более чем два проводника. Ток, текущий к узлу, считается имеющим один знак, от узла — другой. Из первого закона Кирхгофа следует, что алгебраическая сумма токов, сходящихся в узле, равна нулю:

$$\sum_{i=1}^n \mathcal{I}_i = 0.$$

Второй закон Кирхгофа утверждает, что в любом замкнутом контуре, произвольно выбранном в разветвленной электрической цепи, алгебраическая сумма произведений сил токов $\mathcal{I}_i$ на сопротивления R_i соответствующих участков этого контура равна алгебраической сумме приложенных к нему ЭДС:

$$\sum_{i=1}^n \mathcal{I}_i R_i = \sum_{i=1}^m E_i, \quad m \leq n.$$

Используя эти законы для цепи, изображенной на рис. 1, можно записать

$$\begin{aligned} -\mathcal{I}_1 + \mathcal{I}_2 - \mathcal{I}_3 &= 0, \\ R_1 \mathcal{I}_1 + R_2 \mathcal{I}_2 &= E_1 + E_2, \\ R_2 \mathcal{I}_2 + R_3 \mathcal{I}_3 &= E_2. \end{aligned} \quad (1.1)$$

строки и столбцы, получаем так называемую транспонированную матрицу A^T относительно матрицы A :

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}.$$

3. Действия над матрицами. Матрица A с элементами a_{ij} и матрица B с элементами b_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) называются равными, $A = B$, если они имеют одинаковое число строк и столбцов и соответствующие элементы их равны, т. е. $a_{ij} = b_{ij}$.

Суммой двух матриц A и B с одинаковыми размерами $m \times n$ называется матрица C с элементами c_{ij} , которые равны суммам соответствующих элементов a_{ij} и b_{ij} :

$$C = A + B, \quad c_{ij} = a_{ij} + b_{ij}, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n.$$

Из определения суммы матриц непосредственно вытекают следующие ее свойства:

$$A + (B + C) = (A + B) + C,$$

$$A + B = B + A,$$

$$A + 0 = A.$$

Произведение матрицы A на число α или произведение числа α на матрицу A называется матрицей C , элементы которой получены умножением всех элементов матрицы A на число α , т. е.

$$C = A\alpha = \alpha A, \quad c_{ij} = \alpha a_{ij}, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n.$$

Из определения произведения числа на матрицу непосредственно вытекают следующие свойства:

$$1 \cdot A = A,$$

$$0 \cdot A = 0,$$

$$\alpha(\beta A) = (\alpha\beta)A,$$

$$(\alpha + \beta)A = \alpha A + \beta A,$$

$$\alpha(A + B) = \alpha A + \alpha B.$$

Нетрудно убедиться, что если матрицы A и B имеют одинаковые размеры, то

$$A - B = A + (-B).$$

Произведение матрицы A справа на матрицу B , т. е. AB , определено только в том случае, когда число столбцов матрицы A равно числу строк матрицы B . Пусть A — матрица с размерами $m \times n$, а B — матрица с размерами $n \times q$. Произведение этих матриц называется матрицей $C = AB$ с размерами $m \times q$, элементы которой вычисляются по формуле

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj}, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, q.$$

Если оба множителя — квадратные матрицы одинакового порядка, то операция умножения матриц всегда выполнима. Умножение матриц не обладает переместительным законом даже для квадратных матриц. Если $AB = BA$, то матрицы A и B называются перестановочными.

Матричное произведение обладает следующими свойствами:

$$A(BC) = (AB)C, \quad \alpha(AB) = (\alpha A)B,$$

$$(A + B)C = AC + BC, \quad C(A + B) = CA + CB,$$

где A , B и C — матрицы, α — число.

Равенства понимаются в том смысле, что если одна часть существует, то другая часть также существует, и они равны между собой.

Нетрудно показать, что

$$(AB)^T = B^T A^T,$$

аналогично

$$(ABC)^T = C^T B^T A^T$$

и т. д.

Если использовать произведение прямоугольных матриц, то систему линейных алгебраических уравнений (1.4) можно записать одним матричным равенством:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix},$$

или в сокращенном виде

$$Ax = b.$$

4. Определитель квадратной матрицы. Пусть A квадратная матрица с элементами a_{ij} , $i, j = 1, 2, \dots, n$, порядок которой n . Определителем матрицы

$$|A| = \det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

называется сумма $n!$ членов $(-1)^k a_{1k_1} a_{2k_2} \dots a_{nk_n}$, каждый из которых равен произведению n элементов матрицы, взятых по одному из каждой строки и каждого столбца. При этом знак члена определяется количеством инверсий k в перестановке $k_1, k_2, \dots, k_n$ из чисел $1, 2, 3, \dots, n$.

Перестановкой из n чисел $1, 2, 3, \dots, n$ называется любое упорядоченное расположение этих чисел. Количество всех перестановок из n чисел равно $1 \times 2 \times 3 \times \dots \times n = n!$. Например, все перестановки чисел $1, 2, 3$ следующие: 123, 132, 213, 231, 312, 321.

Два числа в перестановке образуют инверсию, если большее число стоит перед меньшим. Например, в перестановке 2413 число инверсий равно $1 + 2 = 3$.

Рассмотрим теперь определитель второго порядка

$$\det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = (-1)^{k(1,2)} a_{11}a_{22} + (-1)^{k(2,1)} a_{12}a_{21} = \\ = (-1)^0 a_{11}a_{22} + (-1)^1 a_{12}a_{21} = a_{11}a_{22} - a_{12}a_{21}.$$

Правило для вычисления определителя второго порядка легко запомнить:

Так же легко получается и запоминается правило для вычисления определителя третьего порядка (для этого к определителю справа приписываются два первых столбца):

$$= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12}.$$

Вычисление определителей более высокого порядка с использованием свойства определителей [55] можно свести к вычислению определителей второго и третьего порядков.

5. Миноры, алгебраические дополнения, ранг матрицы. Обозначим через M_{ij} матрицу порядка $n-1$, полученную вычеркиванием i -й строки и j -го столбца квадратной матрицы A . Тогда $\det M_{ij} = |M_{ij}|$ называется минором элемента a_{ij} матрицы A . Алгебраическим дополнением элемента a_{ij} называется выражение

$$A_{ij} = (-1)^{i+j} |M_{ij}|.$$

Минором k -го порядка матрицы A называется определитель матрицы M , образованной элементами, стоящими на пересечении любых k строк и любых k столбцов матрицы A . Рангом матрицы A называется максимальный порядок отличных от нуля миноров матрицы.

Матрица A имеет ранг $r(A) = r$, если найдется, по крайней мере, один ее минор порядка r , отличный от нуля, а все миноры матрицы A порядка $r+1$ и выше равны нулю.

Для нахождения ранга матрицы теоретически можно поступать следующим образом. Переходить от миноров меньших порядков (на-

чиная с миноров первого порядка, т. е. элементов матрицы) к минорам больших порядков. Если вычислен минор порядка r и он отличен от нуля, то нужно вычислить все миноры порядка $(r+1)$, содержащие этот минор; если все такие миноры порядка $(r+1)$ равны нулю, то ранг матрицы равен r ; если же хотя бы один из них отличен от нуля, то эту операцию можно применить к полученному минору, причем в этом случае ранг матрицы заведомо больше r .

Однако данная рекомендация для определения ранга произвольной матрицы, особенно высокого порядка, на ЭВМ практически нереализуема. Причины этого — большая трудоемкость такого способа и, главное, — невозможность указания надежного критерия равенства нулю вычисленного минора вследствие ограниченной точности вычислений. Для определения ранга матрицы с помощью ЭВМ используются другие методы, о которых речь пойдет дальше.

6. Обратная матрица. Квадратная матрица называется неособенной (невырожденной), если ее определитель отличен от нуля, в противном случае матрица называется особенной (вырожденной).

Для неособенной матрицы важным является понятие обратной матрицы. Обратной матрицей A^{-1} относительно данной матрицы A называется матрица, которая, будучи умноженной как справа, так и слева на данную матрицу A , дает единичную матрицу E :

$$AA^{-1} = A^{-1}A = E.$$

Построение матрицы, обратной относительно данной матрицы, называется обращением данной матрицы.

Необходимым и достаточным условием существования обратной матрицы является неособенность матрицы.

Для матриц небольшого порядка обратную матрицу можно найти следующим образом. Вначале убеждаемся, что $|A| = \det A = \Delta \neq 0$. Затем для матрицы A выписываем так называемую присоединенную матрицу

$$\begin{bmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \dots & \dots & \dots & \dots \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{bmatrix},$$

где A_{ij} — алгебраические дополнения соответствующих элементов a_{ij} , $i, j = 1, 2, \dots, n$ матрицы A .

Далее все элементы присоединенной матрицы разделим на определитель матрицы A , в результате получим

$$A^{-1} = \begin{bmatrix} \frac{A_{11}}{\Delta} & \frac{A_{21}}{\Delta} & \dots & \frac{A_{n1}}{\Delta} \\ \frac{A_{12}}{\Delta} & \frac{A_{22}}{\Delta} & \dots & \frac{A_{n2}}{\Delta} \\ \dots & \dots & \dots & \dots \\ \frac{A_{1n}}{\Delta} & \frac{A_{2n}}{\Delta} & \dots & \frac{A_{nn}}{\Delta} \end{bmatrix}.$$

7. Понятие о собственных значениях и собственных векторах матрицы. Характеристическим уравнением матрицы A с элементами a_{ij} , $i, j = 1, 2, \dots, n$ называется уравнение

$$\det(A - \lambda E) = |A - \lambda E| = \begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{vmatrix} = 0.$$

Корни λ_i характеристического уравнения называются собственными или характеристическими числами матрицы.

Собственным вектором матрицы A , отвечающим собственному значению λ_i , называется ненулевой вектор $\mathbf{v} = [v_1, \dots, v_n]^T$, который удовлетворяет матричному уравнению

$$A\mathbf{v} = \lambda_i \mathbf{v}.$$

т. е. системе уравнений

[illegible]

Задача нахождения собственных значений и собственных векторов представляет большой теоретический и практический интерес.

8. **Специальные матрицы.** Квадратная матрица A называется симметричной, если она равна своей транспонированной матрице: $A = A^T$, т. е. если $a_{ji} = a_{ij}$, $i, j = 1, 2, \dots, n$.

Матрица A называется нормальной, если она перестановочна со своей транспонированной матрицей $AA^T = A^TA$.

Матрица A называется ортогональной, если

$$A^T A = E,$$

т. е. для ортогональной матрицы $A^T = A^{-1}$.

Ортогональная матрица обладает следующими свойствами: строки (столбцы) ортогональной матрицы попарно ортогональны: сумма квадратов элементов каждой строки (столбца) ортогональной матрицы равна 1, определитель ортогональной матрицы равен ± 1 .

Одним из примеров ортогональной матрицы равен ± 1 .
 маая матрица отражения, осуществляющая преобразование произвольного вектора по правилу его отражения от заданной плоскости S . Пусть $\mathbf{w}$ — вектор единичной длины, ортогональный к S . Тогда матрица отражения P относительно S имеет вид

$$P = E - 2\mathbf{w}\mathbf{w}^T.$$

где w — вектор-столбец, w^T — вектор-строка.

Действительно, пусть $\mathbf{z}$ — произвольный вектор, представленный в виде

$$z = x + y,$$

где $\mathbf{x}$ — вектор, ортогональный к $\mathbf{w}$, $(\mathbf{x}, \mathbf{w}) \equiv \mathbf{x}^T \mathbf{w} \equiv \mathbf{w}^T \mathbf{x} = 0$,
а $y = \alpha \mathbf{w}$, α — число. (Заметим, что здесь и в дальнейшем $(\mathbf{u}, \mathbf{v})$ —
скалярное произведение векторов, которое в рассматриваемом случае
вычисляется по формуле $(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v} = \mathbf{v}^T \mathbf{u} = \sum_{i=1}^n u_i v_i$.)

Нетрудно проверить, что

$$Pz = z - 2ww^T x - 2ww^T y = z - 2\alpha w = x - y,$$

т. е. вектор $\mathbf{z}$ преобразуется в вектор, отраженный от плоскости S .

Матрица P — ортогональна, так как

$$PP^T = P^T P = (E - 2\mathbf{w}\mathbf{w}^T)(E - 2\mathbf{w}\mathbf{w}^T) = E - 4\mathbf{w}\mathbf{w}^T + 4\mathbf{w}(\mathbf{w}^T\mathbf{w})\mathbf{w}^T = E.$$

Вектор w всегда можно подобрать так, чтобы матрица P переводила заданный вектор z в вектор заданного направления, например в параллельный какому-нибудь заданному вектору l единичной длины. Для этого достаточно взять

$$w = \frac{1}{\rho} (z - \alpha l),$$

где

$$\alpha = \pm \sqrt{(z, z)}, \quad \rho = \sqrt{2\alpha^2 - 2\alpha(z, l)}.$$

Тогда

$$Pz = z - 2ww^T z = z - \frac{2}{\rho} w [(z, z) - \alpha(l, z)] =$$

$$= z - \frac{2}{|2\alpha^2 - 2\alpha(l, z)|} (z - \alpha l) [\alpha^2 - \alpha(l, z)] = z - (z - \alpha l) = \alpha l.$$

Отметим, что матрица отражения P симметрична.

Симметричная матрица называется положительно определенной, если для всех ненулевых векторов x справедливо неравенство

$$(A\mathbf{x}, \mathbf{x}) \geq 0.$$

Нетрудно убедиться, что все собственные значения положительно определенной матрицы больше нуля.

Для того чтобы симметричная матрица A была положительно определенной, необходимо и достаточно, чтобы выполнялись неравенства

$$a_{11} > 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \quad \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} > 0, \dots, \\ \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} > 0,$$

где $a_{ij} = a_{ji}$, $i, j = 1, 2, \dots, n$.

Симметричная матрица A называется положительно полуопределенной, если при любых вещественных значениях переменных

$$(Ax, x) \geq 0.$$

У положительно полуопределенных матриц A все собственные значения удовлетворяют условию $\lambda_i \geq 0$.

Для того чтобы симметричная матрица A была положительно полуопределенной, необходимо и достаточно, чтобы все ее главные миноры были неотрицательны. Минор матрицы A называется главным, если он образован элементами, стоящими на пересечении строк и столбцов матрицы A с одинаковыми номерами.

Матрица A называется ленточной, если ее элементы удовлетворяют соотношениям

$$a_{ij} = 0, \quad m_1 < i - j; \quad j - i > m_2$$

для некоторых неотрицательных чисел m_1, m_2 .

Если $m_1 = m_2 = m$, то последнее условие можно записать в виде

$$a_{ij} = 0, \quad |i - j| > m.$$

Примером ленточной матрицы может служить матрица

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & a_{35} & 0 & 0 \\ 0 & a_{42} & 0 & a_{44} & a_{45} & a_{46} & 0 \\ 0 & 0 & a_{53} & a_{54} & a_{55} & a_{56} & 0 \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} & a_{67} \\ 0 & 0 & 0 & 0 & 0 & a_{76} & a_{77} \end{bmatrix},$$

где $m = 2$.

Для матрицы

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & a_{24} & 0 & 0 & 0 & 0 & 0 \\ a_{31} & 0 & a_{33} & a_{34} & a_{35} & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & 0 & 0 & 0 \\ a_{51} & 0 & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & 0 & 0 \\ 0 & a_{62} & a_{63} & a_{64} & 0 & a_{66} & a_{67} & a_{68} & 0 \\ 0 & 0 & a_{73} & a_{74} & 0 & a_{76} & a_{77} & 0 & a_{79} \\ 0 & 0 & 0 & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} & a_{89} \\ 0 & 0 & 0 & 0 & a_{95} & a_{96} & 0 & a_{98} & a_{99} \end{bmatrix}$$

число ненулевых диагоналей, расположенных ниже главной диагонали, $m_1 = 4$, число ненулевых диагоналей, расположенных выше главной диагонали, $m_2 = 2$. Шириной ленты будем называть число

$$s = m_1 + m_2 + 1.$$

Квадратная матрица называется треугольной, если ее элементы, стоящие выше (ниже) главной диагонали, равны нулю. Например, матрица

$$T_1 = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ 0 & t_{22} & \dots & t_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & t_{nn} \end{bmatrix},$$

где $t_{ij} = 0$ при $j < i$ есть верхняя треугольная матрица. Аналогично

$$T_2 = \begin{bmatrix} t_{11} & 0 & \dots & 0 \\ t_{21} & t_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ t_{n1} & t_{n2} & \dots & t_{nn} \end{bmatrix}$$

(где $t_{ij} = 0$ для $j > i$) есть нижняя треугольная матрица.

Ряд практических задач описывается системами линейных алгебраических уравнений с профильными матрицами (см., например, матрицу (I.63)). Матрица A n -го порядка называется профильной матрицей, если ее элементы a_{ij} удовлетворяют соотношениям

$$a_{ij} = 0, \quad |i - j| > m_i \geq 0,$$

где m_i в общем случае для каждого i принимает различные значения и почти все они существенно меньше n . В профильных матрицах имеется много нулевых элементов, расположенных не в такой строгой закономерности, как у ленточных матриц. Назовем коэффициентом плотности матрицы отношение количества ненулевых элементов к общему числу ее элементов. Матрицу считают разреженной, если коэффициент ее плотности меньше или равен $1/n$ [95].

9. Нормы векторов и матриц. Нормой вектора $x = [x_1, x_2, \dots, x_n]^T$ называется неотрицательное число $\|x\|$, удовлетворяющее условиям

$$\begin{aligned} \|x\| &\geq 0 && \text{при } x \neq 0, \quad \|0\| = 0, \\ \|Cx\| &= |C| \|x\| && \text{при любом множителе } C, \\ \|x + y\| &\leq \|x\| + \|y\|. \end{aligned}$$

Норма вектора может быть введена различными способами, например следующими:

$$\|x\|_\infty = \max_i |x_i|$$

или

$$\|x\|_1 = \sum_{i=1}^n |x_i|,$$

или

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$$

(евклидова норма).

Нормой квадратной матрицы A называется неотрицательное число $\|A\|$, удовлетворяющее тем же условиям, что и норма векторов, и кроме того условию

Норма матриц может вводиться различными способами. Однако при этом целесообразно, чтобы она была согласована с выбранной нормой векторов, т. е. чтобы для любой матрицы A и любого вектора x выполнялось неравенство $\|Ax\| \leq \|A\| \|x\|$.

$$\|A\| = \max_{\|x\|=1} \|Ax\|,$$

Для введенных ранее норм векторов подчиненными нормами матриц будут следующие:

где μ — наибольшее собственное число матрицы $A^T A$.

Если матрица A симметрична, то $\|A\| = \max_i (\lambda_i)$, где λ_i — собственные значения матрицы A . Иногда эту норму называют спектральной. Сопоставленной с евклидовой нормой векторов является евклидова норма матриц, определяемая выражением

10. Системы линейных алгебраических уравнений. Как было видно, система линейных алгебраических уравнений (I.4) может быть записана так:

Классическим решением системы (1.5) называется упорядоченная совокупность чисел $c_1, c_2, \dots, c_n$, которая (если ее подставить в систему (1.5) вместо неизвестных $x_1, x_2, \dots, x_n$) обращает все уравнения системы в тождества (отметим, что совокупность чисел $c_1, c_2, \dots, c_n$ составляет одно, а не n решений).

Система линейных алгебраических уравнений называется совместной, если она имеет хотя бы одно классическое решение, и несовместной (противоречивой), если она не имеет таких решений.

Совместная система называется определенной, если она имеет единственное решение, и неопределенной, если решений больше одного.

Две системы линейных алгебраических уравнений называются эквивалентными, если каждое решение первой системы является решением второй и каждое решение второй системы является решением первой.

$$B = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{bmatrix}.$$

Вопрос о совместности системы линейных алгебраических уравнений полностью решается теоремой Кронекера — Капелли, которая утверждает, что *система линейных алгебраических уравнений (1.4) совместна тогда и только тогда, когда ранг расширенной матрицы равен рангу матрицы A .*

Совместная система имеет единственное решение, если ранг матрицы системы равен числу неизвестных n , и бесконечно много решений, если $r(A) < n$.

Если матрица A квадратная и $\det A = |A| \neq 0$, то такая система всегда совместна и имеет единственное решение, которое может быть найдено по формулам Крамера

где

Для решения совместной системы линейных алгебраических уравнений (I.4) с прямоугольной матрицей вначале устанавливают ранг матрицы A системы. Затем переставляют уравнения и неизвестные $x_1, x_2, \dots, x_n$ таким образом, чтобы отличный от нуля минор матрицы A , имеющий порядок $r(A)$, занял положение в левом верхнем углу матрицы. При этом могут быть два случая.

В первом случае, когда $r = n$ и $r \leq t$, система (1.4) приобретает вид

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1r}x_r & = & b_1, \\ \cdot & & \cdot \\ a_{r1}x_1 + a_{r2}x_2 + \cdots + a_{mr}x_r & = & b_r, \\ \cdot & & \cdot \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mr}x_r & = & b_m. \end{array}$$

Во втором случае, когда $r < n$, $r \leq m$, систему (I.4) разрешают относительно первых r неизвестных:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1r}x_r &= b_1 - a_{1\ r+1}x_{r+1} - \dots - a_{1\ n}x_n, \\ \dots & \\ a_{r1}x_1 + a_{r2}x_2 + \dots + a_{rr}x_r &= b_r - a_{r\ r+1}x_{r+1} - \dots - a_{r\ n}x_n, \\ \dots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mr}x_r &= b_m - a_{m\ r+1}x_{r+1} - \dots - a_{m\ n}x_n. \end{aligned}$$

Решаем систему из первых r уравнений относительно неизвестных $x_1, x_2, \dots, x_r$. Определитель этой системы отличен от нуля, поэтому она имеет единственное решение. Каждому набору неизвестных $x_{r+1}, x_{r+2}, \dots, x_n$ отвечает единственное решение рассматриваемой системы. Различных наборов значений этих неизвестных может быть бесконечное множество. Поэтому система линейных алгебраических уравнений в этом случае будет неопределенной. При решении систем линейных алгебраических уравнений, коэффициенты и правые части которых заданы точно, анализируют: совместна или несовместна рассматриваемая система линейных алгебраических уравнений; определена или неопределена совместная система уравнений; как описать совокупность решений неопределенной системы; как разработать метод получения единственного решения определенной системы.

1.2. КЛАССИФИКАЦИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ, ОПИСЫВАЮЩИХ ПРИКЛАДНЫЕ ЗАДАЧИ

Рассмотрим, хотя бы теоретически, какие при этом ставятся задачи. Так, в ряде случаев требуется решить систему линейных алгебраических уравнений с невырожденной квадратной матрицей n -го порядка и с одним заданным вектором свободных членов (с одной правой частью) или ту же систему с p правыми частями. В некоторых задачах возникает необходимость найти матрицу, обратную относительно данной невырожденной матрицы порядка n . Существуют задачи, в которых для заданных матрицы A с размерами $m \times n$ и вектора b с m компонентами

2. Системы линейных алгебраических уравнений, описывающих физические модели. Решение прикладных задач, как правило, начинается с создания приемлемых физических и математических моделей. Для их построения используют различные гипотезы. Если эти гипотезы верны (погрешность гипотезы отсутствует или достаточно мала), то физическая модель правильно отражает имеющиеся в прикладной задаче закономерности. Физическая модель может быть описана с помощью математического аппарата, например некоторой системой линейных алгебраических уравнений. При описании физических моделей крайне редко используются системы

$$\tilde{A} \tilde{\mathbf{x}} = \tilde{\mathbf{b}} \quad (\text{I.6})$$

$$Ax = b \quad (1.7)$$
$$\|\tilde{A} - A\| = \|\Delta A\| \leq \varepsilon_A, \quad \|\tilde{\mathbf{b}} - \mathbf{b}\| = \|\Delta \mathbf{b}\| \leq \varepsilon_b. \quad (\text{I.8})$$

Таким образом, для систем линейных алгебраических уравнений с приближенными данными нельзя просто ставить вопрос о «решении индивидуально заданной приближенной системы (1.7)», а следует уточнить, что понимается под решением такого класса задач. В работе [92] указывается, что постановка задачи о решении приближенной системы линейных алгебраических уравнений является фундаментальной задачей вычислительной математики. Эта постановка зависит от доступной информации. Подробнее постановку задач, связанных с решением систем линейных алгебраических уравнений, рассмотрим далее (см. п. 3 настоящего параграфа).

Для удобства точное решение $\tilde{x}$ системы (I.6) с точными, но неизвестными исходными данными будем условно называть физическим решением, а точное решение x заданной системы (I.7) — математическим решением задачи.

Погрешность в решении x , вызванную неточным заданием исходных данных, называют наследственной. Значение ее зависит как от погрешности исходных данных, так и от свойств матрицы.

Решение системы уравнений (I.7), получаемое некоторым численным методом на электронной вычислительной машине, будем называть машинным решением задачи. Вследствие погрешности перевода исходных данных из десятичной системы в двоичную, погрешности метода и погрешности машинной реализации алгоритма машинное решение задачи может отличаться от математического.

Таким образом, при решении систем линейных алгебраических уравнений, описывающих прикладные задачи, необходимо определить понятие искомого решения, построить устойчивый алгоритм нахождения этого решения, оценить в ходе решения задачи погрешность машинной реализации (близость машинного и математического решений), а также оценить наследственную погрешность (близость математического и физического решений).

Для получения достоверного решения задачи принципиально важным является определение специфических свойств задачи.

3. Корректно и некорректно поставленные задачи. Рассмотрим постановки задач о решении системы линейных алгебраических уравнений (I.7), (I.8) в случае прямоугольной матрицы A с размерами $m \times n$ и произвольного вектора правых частей.

Как известно, задача является корректно поставленной, если решение ее существует и единственно при любых входных данных из некоторой области их изменения и это решение непрерывно зависит от исходных данных.

Отыскание классического решения x системы (I.7), (I.8), т. е. вектора x , для которого невязка $g = b - Ax$ тождественно равна нулю, будет корректно поставленной задачей, если $m = n$, $\det A \neq 0$ и

$$\| \Delta A A^{-1} \| < 1 \quad \text{или} \quad \| \Delta A \| \| A^{-1} \| < 1 \quad (\text{I.9})$$

при произвольном возмущении ΔA из окрестности, определяемой условиями (I.8).

Действительно, выполнение условия (I.9) гарантирует, что при любом возмущении элементов в пределах точности их задания матрица системы останется невырожденной, а следовательно, решение $x = A^{-1}b$ заданной системы существует, единственно, и оно непрерывно зависит от исходных данных. Для пояснения последнего утверждения рассмотрим разность $x' - x$ решений системы (I.7) и произвольной возмущенной системы

$$A'x' = b',$$

где

$$A' = A + \Delta A, \quad b' = b + \Delta b, \quad x' = x + \Delta x.$$

Легко проверить, что при условии (I.9)

$$x' - x = A^{-1} (E + \Delta A A^{-1})^{-1} (\Delta b - \Delta A A^{-1} b).$$

Следовательно,

$$\| x' - x \| \leq \frac{\| A^{-1} \|}{1 - \| \Delta A A^{-1} \|} (\| \Delta b \| + \| \Delta A \| \| x \|),$$

или

$$\| x' - x \| \leq \frac{\| A^{-1} \|}{1 - \| \Delta A A^{-1} \|} (\| \Delta b \| + \| \Delta A A^{-1} \| \| b \|),$$

откуда очевидна непрерывная зависимость решения системы (I.7) — (I.9) от входных данных задачи.

Например, задача отыскания классического решения системы

$$25x_1 - 36x_2 = 1, \quad 16x_1 - 23x_2 = -1 \quad (\text{I.10})$$

с приближенными исходными данными является корректно поставленной, если погрешность задания элементов матрицы будет удовлетворять условию $\| \Delta A \|_{\infty} \leq 0,015$. Действительно, поскольку

$$A^{-1} = \begin{bmatrix} -23 & 36 \\ -16 & 25 \end{bmatrix}, \quad \| A^{-1} \|_{\infty} = 59,$$

то условие (I.9) выполняется, а следовательно, гарантируются невырожденность матрицы при любых допустимых возмущениях исходных данных и непрерывная зависимость решения от этих данных.

Для корректной постановки задачи об отыскании решения системы линейных алгебраических уравнений (I.7) с произвольной и, в частности, с прямоугольной матрицей необходимо уточнить понятие искомого решения.

Обобщенным решением в смысле наименьших квадратов системы (I.7) называется любой вектор x_0 , для которого евклидова норма невязки достигает наименьшего значения:

$$\min_x \| b - Ax \|_E^2 = \| b - Ax_0 \|_E^2.$$

Решение x_H , имеющее наименьшую евклидову норму

$$\min_{x_0} \| x_0 \|_E = \| x_H \|,$$

называется нормальным обобщенным решением.

Решение x_H можно определить и как обобщенное решение, ортогональное всем линейно независимым классическим решениям однородной системы

$$Av = 0,$$

т. е. $(x_H, v_j) \equiv x_H^T v_j = 0, j = 1, 2, \dots, k$. Нормальное обобщенное решение всегда существует и единственно. Очевидно, что нормальное обобщенное решение будет и классическим в случае $m = n, \det A \neq 0$.

Легко убедиться, что обобщенные решения и только они являются классическими решениями системы

$$A^T A x = A^T b. \quad (I.11)$$

Задача отыскания определенного выше нормального обобщенного решения x_H системы (I.7) корректна, если A является матрицей полного ранга, т. е. $r_A = \min(m, n)$, и изменение этой матрицы в пределах точности ее задания (I.8) не вызовет изменения ранга. Последнее условие выполняется, если норма возмущения $\|\Delta A\|$ мала по сравнению с минимальным ненулевым сингулярным числом матрицы A . Сингулярными числами матрицы A называют положительные квадратные корни из общих собственных значений матриц $A^T A$ и $A A^T$.

Как известно, нормальное обобщенное решение системы (I.7) с матрицей полного ранга при $m \geq n$ является единственным обобщенным решением этой системы и единственным классическим решением системы (I.11) с квадратной невырожденной матрицей $A^T A$ порядка n .

Нормальное обобщенное решение x_H системы (I.7) с матрицей полного ранга при $m < n$ можно получить, например, из единственного классического решения системы

$$A A^T y = b \quad (I.12)$$

с квадратной невырожденной матрицей $A A^T$ порядка m путем преобразования $x_H = A^T y$.

Для систем с прямоугольной матрицей неполного ранга или с квадратной вырожденной постановка задач об отыскании решения требует дальнейшего уточнения. Это объясняется тем, что в упомянутых случаях как угодно малые возмущения элементов матрицы могут вызывать изменение ранга, а следовательно, скачкообразные (не непрерывные) изменения решения. Ранее были рассмотрены некоторые некорректно поставленные задачи [91, 93, 94] и разработаны специальные методы их решения [93, 94, 105], которые будут описаны нами в гл. IV.

4. Классификация корректно поставленных задач. Корректно поставленные задачи решения систем линейных алгебраических уравнений в зависимости от чувствительности решения к погрешности в исходных данных можно разделить на хорошо и плохо обусловленные.

Приведем некоторые примеры, характеризующие зависимость наследственной погрешности от свойств матрицы и погрешности в исходных данных. Так, в системах

$$\begin{aligned} 2x_1 - x_2 &= 0, & 2x_1 - x_2 &= 0, \\ -x_1 + 2x_2 &= 3, & -x_1 + 2x_2 &= 3,003 \end{aligned}$$

свободные члены различаются между собой в третьем десятичном знаке. Решением первой из них является $x_1 = 1$, $x_2 = 2$, а второй — $x_1 = 1,001$, $x_2 = 2,002$. Погрешность в третьем десятичном знаке правой части последней системы вызывает изменение решения тоже в третьем десятичном знаке и не приводит к существенному искажению получаемого решения.

Свободные члены систем уравнений

$$\begin{aligned} 100x_1 + 500x_2 &= 1700, & 100x_1 + 500x_2 &= 1700, \\ 15x_1 + 75,01x_2 &= 255 & 15x_1 + 75,01x_2 &= 255,03 \end{aligned} \quad (I.13)$$

различаются в пятой значащей цифре. Решением первой из них является $x_1 = 17$, $x_2 = 0$, а второй $x_1 = 2$, $x_2 = 3$. Отметим, что определитель этих систем равен единице.

Таким образом, при решении корректно поставленных задач наряду с получением единственного классического решения задачи возникает необходимость в оценке наследственной погрешности или близости математического и физического решений задачи.

Верхнюю границу «относительной» наследственной погрешности точного решения x системы (I.7) можно выразить через «относительные» погрешности заданных матрицы A и вектора b следующим образом [17]:

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \frac{\|A\| \|A^{-1}\|}{1 - \|\Delta A\| \|A^{-1}\|} \left[\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right], \quad (I.14)$$

или

$$\frac{\|x - \tilde{x}\|}{\|\tilde{x}\|} \leq \frac{\|A\| \|A^{-1}\|}{1 - \frac{\|\Delta b\|}{\|b\|}} \left[\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right], \quad (I.15)$$

при условии $\|\Delta A\| \|A^{-1}\| < 1$ и естественном предположении $\frac{\|\Delta b\|}{\|b\|} < 1$. Обе оценки обычно являются мажорантными. Однако всегда можно построить пример, когда указанная граница достигается [77], т. е. оценки (I.14), (I.15) являются неулучшаемыми на всем классе невырожденных матриц.

Из формул (I.14), (I.15) очевидно, что устойчивость решения системы к изменениям исходных данных в значительной степени зависит от величины $\text{cond } A = \|A\| \|A^{-1}\|$, которая называется числом обусловленности матрицы. Если $\text{cond } A$ невелико, то матрица A системы линейных алгебраических уравнений называется хорошо обусловленной. Если $\text{cond } A$ велико, то матрица такой системы называется плохо обусловленной. В зависимости от способов введения норм матрицы рассматривают несколько видов чисел обусловленности матриц. Например, для симметричных матриц

$$\text{cond } A = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|},$$

где λ_i , $i = 1, 2, \dots, n$, — собственные значения матрицы A , для несимметричных матриц

$$\text{cond } A = \sqrt{\frac{\mu_n}{\mu_1}}.$$

Здесь μ_n и μ_1 — наибольшее и наименьшее собственные значения матрицы $A^T A$. Некоторые другие числа обусловленности рассматривались в работе [103].

Из оценок (I.14), (I.15) следует, что достоверность полученного математического решения определяется не только обусловленностью матрицы системы линейных алгебраических уравнений, но и точностью задания исходных данных. Величину

$$m = \text{cond } A \left[\frac{\| \Delta A \|}{\| A \|} + \frac{\| \Delta b \|}{\| b \|} \right] \quad (\text{I.16})$$

будем называть числом обусловленности системы линейных алгебраических уравнений. Формула (I.16) увязывает свойства матрицы системы и погрешность в задании исходных данных. В реальных задачах есть смысл рассматривать те системы, для которых оценки числа m заметно меньше единицы, например $m \leq 0,01$.

Таким образом, рассмотрение устойчивости решения к изменению исходных данных для систем линейных алгебраических уравнений с квадратной невырожденной матрицей A , удовлетворяющей условию (I.9), позволяет выделить хорошо и плохо обусловленные системы.

Из формулы (I.16) следует, что для уменьшения наследственной погрешности математического решения необходимо стремиться к уменьшению числа m либо за счет увеличения точности задания исходных данных, либо за счет переформулировки изучаемой физической модели относительно других параметров с целью уменьшения числа обусловленности матрицы.

Так, вводя погрешность в шестой десятичный знак свободного члена первой системы (I.13), а именно рассматривая систему

$$100x_1 + 500x_2 = 1700,$$

$$15x_1 + 75,01x_2 = 255,000\,003,$$

получаем $x_1 = 16,9985$, $x_2 = 0,000\,3$, что является достаточно хорошим приближением к решению $x_1 = 17$, $x_2 = 0$ невозмущенной первой системы (I.13).

Если нельзя повысить точность задания исходных данных или переформулировать практическую задачу так, чтобы получить достаточно хорошо обусловленную систему, то для плохо обусловленной системы иногда разыскивают устойчивую проекцию решения на подпространство, образованное собственными векторами матрицы A^*A , отвечающими «большим» собственным значениям. В целом, вопрос о решении плохо обусловленных систем линейных алгебраических уравнений до конца не решен.

Из рассмотренного выше следует, что до решения прикладной задачи на ЭВМ может сложиться ситуация, при которой точное решение полученной математической задачи может не быть решением рассматриваемой нами физической модели. Реализация численного метода на ЭВМ вносит свою погрешность, которую необходимо учитывать при анализе машинного решения задачи.

5. Погрешность реализации вычислительных алгоритмов на ЭВМ. Здесь и в дальнейшем при анализе машинной реализации алгоритмов вычислительной математики будем иметь в виду некоторую абстрактную вычислительную машину, которая производит вычисления в двоичной системе числения в режиме с плавающей запятой. При вычислениях с плавающей запятой каждое стандартное число x представимо упоря-

доченной парой чисел: b (число, называемое порядком) и a (число, называемое мантиссой) так, что $x = 2^b a$; b — целое положительное или отрицательное число, a удовлетворяет неравенствам

$$-\frac{1}{2} \geq a > -1, \quad \frac{1}{2} \leq a < 1.$$

Будем предполагать, что a имеет t цифр после двоичной запятой и будем считать, что вычислительная машина имеет « t -разрядную мантиссу». При введении почти всех чисел в ЭВМ вносится некоторая ошибка, связанная с их округлением и стандартной формой представления, характерной для каждой конкретной ЭВМ. Обозначим через $\text{fl}(x)$ конечную дробь, которая получается после округления мантиссы числа до t -го знака после запятой. Тогда

$$\text{fl}(x) \equiv x(1 + \varepsilon), \quad |\varepsilon| \leq 2^{-t}.$$

Допустим, что ноль имеет нестандартное представление, в котором

$$a = b = 0.$$

Равенство вида

$$z = \text{fl} \left(x \underset{i}{\times} y \right)$$

будет обозначать, что x , y и z — стандартные числа с плавающей запятой и что z получено из x и y выполнением соответствующей операции с плавающей запятой. Предположим, что ошибки округления в этих операциях таковы:

$$z = \text{fl} \left(x \underset{i}{\times} y \right) = \left(x \underset{i}{\times} y \right) (1 + \varepsilon),$$

где

$$|\varepsilon| \leq 2^{-t},$$

поэтому каждая из конкретных арифметических операций дает результат, имеющий незначительную ошибку округления.

Влияние ошибок машинной реализации рассмотрим на следующем простом примере. Пусть на ЭВМ серии МИР¹ требуется найти решение системы линейных алгебраических уравнений

$$\begin{aligned} 0,135x_1 + 0,188x_2 + 0,191x_3 + 0,178x_4 &= 0,3516, \\ 0,188x_1 + 0,262x_2 + 0,265x_3 + 0,247x_4 &= 0,4887, \\ 0,191x_1 + 0,265x_2 + 0,281x_3 + 0,266x_4 &= 0,5105, \\ 0,178x_1 + 0,247x_2 + 0,266x_3 + 0,255x_4 &= 0,4818. \end{aligned} \quad (\text{I.17})$$

Учитывая, что в математической задаче (I.17) правая часть имеет четыре десятичных знака, для решения такой системы одним из прямых

¹ У ЭВМ серии МИР, предназначенных для инженерных расчетов, длина машинного слова в десятичных знаках в принципе может быть произвольной (в пределах возможностей оперативной памяти).

методов может быть интуитивно выбрана длина машинного слова в шесть десятичных знаков. В результате решения системы (I.17) методом LL^T разложения (метод квадратных корней) получаем машинное решение

$$x^{(1)} = \begin{bmatrix} -0,0378848 \\ 0,764402 \\ 0,710031 \\ 0,434482 \end{bmatrix}.$$

Вычисление невязки $r^{(1)} = b - Ax^{(1)}$ с длиной машинного слова в двенадцать десятичных знаков дает

$$r^{(1)} = \begin{bmatrix} -0,245 \cdot 10^{-6} \\ -0,3506 \cdot 10^{-6} \\ -0,12562 \cdot 10^{-5} \\ -0,14556 \cdot 16^{-5} \end{bmatrix}.$$

Однако, как нетрудно убедиться, математическое решение системы (I.17) есть

$$x = \begin{bmatrix} 0,4 \\ 0,5 \\ 0,6 \\ 0,5 \end{bmatrix}. \quad (I.18)$$

Из этого примера видно, что машинное решение задачи может сильно отличаться от ее математического решения. Рассмотрим некоторые причины такого явления.

Одним из источников погрешности в $x^{(1)}$ являются ошибки округления, возникающие при вводе информации в ЭВМ (погрешность перевода из одной системы счисления в другую, использование элементарных функций, которые в ЭВМ вычисляются по приближенным формулам и т. д.). Например, в результате ввода в ЭВМ ЕС с одинарной точностью элементов матрицы A и правой части b системы (I.17) в памяти машины будут храниться следующие массивы данных в шестнадцатичной системе счисления:

$$\bar{A} = \begin{bmatrix} 402285C & 403020C4 & 4030E560 & 4029168 \\ 403020C4 & 4043126E & 404370A & 4033B64 \\ 4030E560 & 404370A & 4047E9 & 40441893 \\ 4029168 & 4033B64 & 40441893 & 404147AE \end{bmatrix},$$

$$\bar{b} = [405A0275 \quad 4071B71 \quad 4082B020 \quad 407B573E]^T.$$

Заметим, что здесь для представления шестнадцатичных цифр используются следующие обозначения:

$$A = 10, B = 11, C = 12, D = 13, E = 14.$$

Далее первые две цифры каждого из приведенных шестнадцатичных чисел обозначают, как принято в ЭВМ ЕС, «смещенный» порядок, т. е. $40 = 16 \cdot 4 + 0 = 64$, что соответствует нулевому «несмещенному» порядку числа с плавающей запятой (подробнее о представлении чисел в ЭВМ ЕС см. [35]).

Массивам $\bar{A}$ и $\bar{b}$ шестнадцатичных чисел соответствуют (с точностью до восьми десятичных значащих цифр) следующие массивы десятичных чисел:

$$\begin{bmatrix} 0,134\,999\,99 & 0,187\,999\,96 & 0,190\,999\,98 & 0,177\,999\,97 \\ 0,187\,999\,96 & 0,261\,999\,96 & 0,264\,999\,99 & 0,246\,999\,98 \\ 0,190\,999\,98 & 0,264\,999\,99 & 0,280\,999\,96 & 0,265\,999\,97 \\ 0,177\,999\,97 & 0,246\,999\,98 & 0,265\,999\,97 & 0,255\,000\,00 \end{bmatrix},$$

$$\begin{bmatrix} 0,351\,599\,99 \\ 0,488\,699\,97 \\ 0,510\,499\,95 \\ 0,481\,799\,96 \end{bmatrix}.$$

Таким образом, после ввода матрицы и вектора правых частей системы в ЭВМ нужно рассматривать уже не систему (I.17), а некоторое возмущенное уравнение

$$\bar{A}\bar{x} = \bar{b}, \quad (I.19)$$

которое будем называть машинной задачей.

Для решения систем линейных алгебраических уравнений применяются прямые и итерационные методы. Прямые методы характеризуются тем, что дают решение системы за конечное число арифметических операций. Если все операции выполняются точно (без ошибок округления), то решение заданной системы тоже получается точным. Итерационные методы являются приближенными. Они дают решение системы как предел последовательных приближений, вычисляемых некоторым единообразным процессом. При использовании итерационного метода всегда предполагается, что на практике строится не бесконечная последовательность приближений к решению, а процесс останавливается на каком-то s -м приближении, поэтому в машинном решении может быть некоторая погрешность, называемая погрешностью метода решения.

Источником погрешности в машинном решении являются и ошибки, возникающие при реализации вычислительных алгоритмов на ЭВМ, так как все вычисления выполняются с округлением результатов арифметических действий до некоторого фиксированного количества разрядов у мантиисы чисел.

Рассмотрим пример, иллюстрирующий влияние ошибок округления. Как отмечено выше, решая при одинарной точности ЭВМ ЕС систему (I.17), в действительности будем решать систему (I.19). И если точное решение системы (I.17) имеет вид (I.18), то точное решение системы (I.19) в пределах семи значащих цифр

$$\bar{x} = [0,517\,405\,0, \quad 0,429\,092\,4, \quad 0,570\,514\,7, \quad 0,517\,487\,0]^T.$$

Таким образом, вследствие ошибок округления, возникающих при вводе чисел в ЭВМ, точные решения имеющейся системы (I.17) и машинной задачи (I.19) могут весьма сильно различаться.

Далее, решая систему (I.17) (в действительности систему (I.19)) на ЭВМ ЕС методом квадратных корней с одинарной точностью, вследствие накопления ошибок округления при арифметических операциях получаем результат

$$x^{(1)} = [0,408\ 930\ 72, 0,494\ 605\ 78, 0,597\ 760\ 26, 0,501\ 327\ 40]^T,$$

который весьма сильно отличается от точного решения $\bar{x}$ системы (I.19).

Если же вычисление системы (I.19) выполним с удвоенной точностью, то тем же методом квадратных корней получим результат, полностью совпадающий с $\bar{x}$ в пределах семи значащих цифр, но не совпадающий с точным решением (I.18) системы (I.17).

Если и ввод данных системы (I.17), и вычисление решения методом квадратных корней осуществлять с удвоенной точностью ЭВМ ЕС, то получаем (в пределах десяти значащих цифр):

$$x = [0,400\ 000\ 000\ 1, 0,499\ 999\ 999\ 9, 0,600\ 000\ 000\ 0, 0,500\ 000\ 000\ 0]^T,$$

что хорошо согласуется с (I.18).

Следуя работе [97], будем считать, что суммарное влияние ошибок округления в прямых методах можно рассматривать как соответствующее эквивалентное возмущение исходных данных.

Поэтому вычисленное машинное решение $x^{(1)}$ системы (I.7) является точным для некоторой возмущенной системы, например

$$(A + dA)x^{(1)} = b + db, \quad (I.20)$$

или

$$(A + F)x^{(1)} = b, \quad (I.21)$$

и приближенным, не совпадающим с математическим, решением системы (I.7). Здесь dA , db , F — соответствующие эквивалентные возмущения, зависящие от метода решения, порядка системы, длины мантиссы t машинного слова и от погрешностей перевода из десятичной в машинную системы счисления.

Справедливы следующие оценки:

$$\frac{\|x^{(1)} - x\|}{\|x^{(1)}\|} \leq \frac{\|A\| \|A^{-1}\|}{1 - \frac{\|dA\|}{\|A\|}} \left[\frac{\|dA\|}{\|A\|} + \frac{\|db\|}{\|b\|} \right], \quad (I.22)$$

$$\frac{\|x^{(1)} - x\|}{\|x\|} \leq \frac{\|A\| \|A^{-1}\|}{1 - \|dA\| \|A\|} \left[\frac{\|dA\|}{\|A\|} + \frac{\|db\|}{\|b\|} \right], \quad (I.23)$$

$$\frac{\|x^{(1)} - x\|}{\|x^{(1)}\|} \leq \|F\| \|A^{-1}\|. \quad (I.24)$$

Оценки (I.22) — (I.24) свидетельствуют о том, что ошибки машинной реализации особенно опасны для систем линейных алгебраических уравнений с большими числами обусловленности.

Рассмотрение оценок (I.22) — (I.24) позволяет сделать вывод о хорошей или плохой обусловленности системы линейных алгебраических уравнений в связи с численной устойчивостью машинного решения к суммарным ошибкам округления. Но здесь понятие хорошей или плохой «машинной обусловленности» системы тесно связано с возможностями конкретной машины.

В результате одна и та же система может классифицироваться для одной машины или одной длины мантиссы машинного слова как «машинно плохо обусловленная», а для другой — как «машинно хорошо обусловленная». Увеличивая длину машинного слова, всегда можно получить машинное решение, достаточно близкое к математическому решению задачи.

Необходимо отметить два принципиальных момента, касающихся увеличения длины машинного слова при решении систем линейных алгебраических уравнений. Во-первых, для систем с невырожденными матрицами, для которых выполнено условие (I.9), никакое повышение разрядности вычислений не может приблизить математическое решение

x к физическому решению $\bar{x}$ задачи. Во-вторых, для некорректно поставленных задач изменение длины машинного слова не может помочь в построении решения обычными методами, даже если исходные данные системы будут точными, а сама система совместна. В подобных задачах не существует «классического» единственного решения. При рассмотрении таких задач необходимо доопределить понятие искомого решения и использовать специальные алгоритмы его построения.

Численное решение на ЭВМ систем линейных алгебраических уравнений, описывающих прикладные задачи, является нетривиальной задачей и не всегда приводит нас к желаемому результату. Для оценки достоверности получаемых решений необходимо иметь информацию как о рассматриваемой системе уравнений, так и о математических характеристиках ЭВМ.

В основе анализа достоверности решения задач лежит правильное определение их специфических свойств. Если прикладная задача, точнее ее физическая модель, описывается системой линейных алгебраических уравнений, то необходимо выяснить, корректно или некорректно поставлена эта задача. Если задача корректно поставлена, то необходимо определить, хорошо или плохо обусловлена рассматриваемая система. Ответы на эти вопросы и сопоставление числа обусловленности матрицы с погрешностью в исходных данных позволяют оценить близость математического и физического решений задачи.

В ходе машинной реализации алгоритма целесообразно исследовать машинную обусловленность системы (I.19), влияние погрешности машинной реализации и погрешности в машинных исходных данных на достоверность получаемого решения, т. е. оценивать близость машинного и математического решений задачи.

Такой анализ систем уравнений обусловлен постановкой задач, отличной от классической, при описании прикладных проблем и машинной реализацией алгоритмов. До сих пор нами обсуждались лишь теоретические аспекты достоверности получаемых на машинах решений.

В дальнейшем рассмотрим некоторые практические приемы, позволяющие оценить близость машинного к математическому, а математического к физическому решениям.

1.3. НЕКОТОРЫЕ ПРЯМЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ С НЕВЫРОЖДЕННЫМИ МАТРИЦАМИ

1. Предварительные сведения. Для решения систем линейных алгебраических уравнений, как известно, применяют прямые и итерационные методы решения. В прямых методах число необходимых для решения задачи арифметических операций зависит только от вида вычислительной схемы и от порядка матрицы.

Прямые методы удобно применять для решения систем линейных алгебраических уравнений с плотными или разреженными (со специальным способом хранения, исключающим хранение большинства нулей) матрицами, целиком помещающимися в оперативную память ЭВМ. Большинство прямых методов основано на идее последовательного эквивалентного преобразования заданной системы с целью исключения неизвестных из части уравнений. В результате исходная система линейных алгебраических уравнений (1.7) порядка n преобразуется в эквивалентную ей систему

$$A^{(n-1)}\mathbf{x} = \mathbf{b}^{(n-1)} \quad (A^{(n)}\mathbf{x} = \mathbf{b}^{(n)}) \quad (\text{I.25})$$

с матрицей $A^{(n-1)}$ более простой формы, например треугольной или диагональной. Таким образом, решение эквивалентной системы (1.25) уже не представляет труда. Процесс эквивалентного преобразования можно представить как последовательные умножения матрицы A и вектора b слева на матрицы M_i , $i = 1, 2, \dots, n - 1$, причем в результате одного умножения аннулируется, по крайней мере, один элемент преобразуемой матрицы. В результате получают

$$MA\mathbf{x} = U\mathbf{x} \equiv A^{(n-1)}\mathbf{x} = \mathbf{b}^{(n-1)} \equiv M\mathbf{b}, \quad (\text{I.26})$$

$$M = M_{n-1}M_{n-2} \dots M_1.$$

Матрицы M_i могут быть различной формы: треугольные, ортогональные и т. д. Различные модификации методов исключения по существу являются методами разложения матрицы A в произведение двух треугольных или треугольной и ортогональной матриц. Действительно, согласно (1.26) можно записать $A = LA^{(n-1)}$, где $L = M^{-1}$, т. е. $L = M_1^{-1}M_2^{-2} \dots M_{n-1}^{-1}$, откуда следует, что $A = LU$. Алгебраической основой гауссового исключения является следующее утверждение [109].

Пусть дана квадратная матрица A порядка n и A_k означает главный минор матрицы, составленный из первых k строк и столбцов. Предположим, что $\det A_k \neq 0, k = 1, 2, \dots, n-1$. Тогда существуют единственная нижняя треугольная матрица L с единицами на главной диагонали и единственная верхняя треугольная матрица U такие, что $LU = A$ и $\det A = u_{11}u_{22}\dots u_{nn}$.

Из утверждения следует, что симметричная положительно определенная матрица имеет единственное разложение вида LL^T , где L — нижняя треугольная матрица с положительными диагональными элементами.

Использование того или иного варианта прямого метода решения системы линейных алгебраических уравнений с невырожденной матрицей зависит как от свойств матрицы системы (симметричной или несимметричной, порядка ее, специфики вида матрицы, степени разреженности и т. д.), так и технической и математической характеристик ЭВМ и ее операционной системы (длины машинного слова, возможности аппаратного или программного проведения вычислений с удвоенной точностью, режима фиксированной или плавающей запятой, быстродействия ЭВМ как по арифметическим, так и по логическим операциям, объема и организации ее памяти).

2. Метод Гаусса. Если матрица системы линейных алгебраических уравнений несимметрична и электронная вычислительная машина не имеет аппаратной реализации вычислений с удвоенной точностью, а программная реализация этих вычислений снижает быстродействие в несколько раз, то в этих условиях целесообразно использовать метод Гаусса с выбором главного элемента, который будет рассмотрен ниже.

Метод Гаусса (схема единственного деления) для решения системы линейных алгебраических уравнений

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ \vdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \quad (1.27)$$

с матрицей A и правой частью b вида

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad (1.28)$$

целиком помещающимися в оперативной памяти ЭВМ, может быть реализован следующим образом. Предположим, что $a_{11} \neq 0$, разделим первое уравнение системы (1.27) на коэффициент a_{11} , который называют ведущим для первого шага. Затем полученное уравнение умножаем последовательно на a_{i1} , $i = 2, 3, \dots, n$, и вычитаем его из оставшихся уравнений $i = 2, 3, \dots, n$ системы (1.27). В результате вместо системы (1.27) получаем эквивалентную ей систему

[illegible]

с матрицей

$$A^{(1)} = \begin{bmatrix} 1 & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{bmatrix}.$$

Отметим, что матрицу $A^{(1)}$ можно получить из $A^{(0)} = A$ умножением слева на матрицу M_1 вида

$$M_1 = \begin{bmatrix} \frac{1}{a_{11}} & 0 & 0 & \dots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{11}} & 0 & 0 & \dots & 1 \end{bmatrix}, L_1 = M_1^{-1} = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & 0 & 0 & \dots & 1 \end{bmatrix}.$$

С системой (I.29) поступаем так же, как с системой (I.27), исключая x_2 из всех уравнений кроме первого и второго. Аналогично M_1 можно выписать матрицу M_2 и $L_2 = M_2^{-1}$. На k -м шаге такого преобразования k -я строка матрицы $A^{(k-1)}$ делится на ее элемент $a_{kk}^{(k-1)}$, умножается последовательно на $a_{ik}^{(k-1)}$, $i = k + 1, \dots, n$, и вычитается из всех следующих за ней строк так, чтобы все ненулевые элементы k -го столбца, лежащие ниже k -й строки, становились нулями. Отметим, что

$$A^{(k)} = M_k A^{(k-1)}, \quad \mathbf{b}^{(k)} = M_k \mathbf{b}^{(k-1)}, \quad k = 1, 2, \dots, n, \quad (\text{I.30})$$

элементы k -го столбца нижней треугольной матрицы M_k выражаются формулами

$$m_i^{(k)} = 0, \quad i < k, \quad m_k^{(k)} = \frac{1}{a_{kk}^{(k-1)}}, \quad m_i^{(k)} = -\frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}}, \quad i > k. \quad (1.31)$$

Отметим, что $L_k = M_k^{-1}$.

Нижняя треугольная матрица M_k (рис. 2) может быть представлена в виде

$$M_k = E + (M^{(k)} - \mathbf{e}_k) \mathbf{e}_k^\top, \quad (\text{I.32})$$

где e_k есть k -й столбец единичной матрицы E n -го порядка.

Из (1.30) получаем

$$U = A^{(n)} = M_n A^{(n-1)} = M_n M_{n-1} M_{n-2} \dots M_1 A^{(0)}.$$

Введя обозначения

$$M = M_n M_{n-1} M_{n-2} \dots M_1, \quad L = L_1 L_2 \dots L_{n-1} L_n, \quad (\text{I.33})$$

можно записать

$$MA = U, \quad (\text{I.34})$$

$$A = LU. \quad (1.35)$$

$$U_k = \begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & 1 & & & & & & \\ & & & & X & & & & & \\ & & & & \vdots & & & & & \\ & & & & & X & & & & \\ & & & & & & X & & & \\ & & & & & & & 1 & & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{bmatrix}_k$$

Рис. 2.

Таким образом, после n -го шага приходят к следующей системе уравнений:

[illegible]

причем

$$U = A^{(n)} = \begin{bmatrix} 1 & a_{12}^{(1)} & a_{13}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & 1 & a_{23}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix},$$

$$L = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22}^{(1)} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2}^{(1)} & a_{n3}^{(2)} & \dots & a_{nn}^{(n-1)} \end{bmatrix}. \quad (I.37)$$

Эту систему в матричном виде можно записать так:

$$U\mathbf{x} = M\mathbf{b}. \quad (\text{I.38})$$

Преобразование системы (I.27) в эквивалентную ей систему (I.36) называют прямым ходом, а решение треугольной системы (I.36) — обратным.

Вычислительные формулы схемы единственного деления метода Гаусса имеют следующий вид. Прямой ход, s -й шаг $s = 1, 2, \dots, n$:

$$a_{ik}^{(s)} = \frac{a_{ik}^{(s-1)}}{a_{ss}^{(s-1)}}, \quad b_i^{(s)} = \frac{b_i^{(s-1)}}{a_{ss}^{(s-1)}}, \quad i = s, \quad k = s, \quad s + 1, \dots, n,$$

$$a_{ik}^{(s)} = a_{ik}^{(s-1)} - a_{sk}^{(s)} a_{is}^{(s-1)}, \quad b_i^{(s)} = b_i^{(s-1)} - b_s^{(s)} a_{is}^{(s-1)},$$

$$i = s + 1, s + 2, \dots, n; \quad k = s, s + 1, \dots, n.$$

Обратный ход реализуется по формуле

$$x_i = b_i^{(i)} - \sum_{k=i+1}^n a_{ik}^{(i)} x_k, \quad i = n, n-1, \dots, 1.$$

Из разложений (I.35), (I.37) следует

$$\det A = \det L \det U = \prod_{i=1}^n a_{ii}^{(i-1)}, \quad a_{11} = a_{11}^{(0)}.$$

Это свойство используется на практике для вычисления определителя матрицы A методом Гаусса.

Пример 1. По описанной схеме метода Гаусса решить систему линейных алгебраических уравнений

$$\begin{aligned} 5x_1 + x_2 + 2x_3 &= 8, \\ 4x_1 + 4x_2 + x_3 &= 2, \\ x_1 - 2x_2 + 3x_3 &= 9. \end{aligned} \quad (I.39)$$

Решение. После первого шага получаем эквивалентную (I.39) систему

$$\begin{aligned} x_1 + 0,2x_2 + 0,4x_3 &= 1,6, \\ 3,2x_2 - 0,6x_3 &= -4,4, \\ -2,2x_2 + 2,6x_3 &= 7,4. \end{aligned}$$

В результате второго шага имеем

$$\begin{aligned} x_1 + 0,2x_2 + 0,4x_3 &= 1,6, \\ x_2 - 0,1875x_3 &= -1,375, \\ 2,1875x_3 &= 4,375. \end{aligned}$$

Прямой ход завершается получением системы вида

$$\begin{aligned} x_1 + 0,2x_2 + 0,4x_3 &= 1,6, \\ x_2 - 0,1875x_3 &= -1,375, \\ x_3 &= 2. \end{aligned}$$

В результате обратного хода находим

$$x_3 = 2, \quad x_2 = -1,375 + 0,1875 \cdot 2 = -1, \quad x_1 = 1$$

Пример 2. Найти, используя схему единственного деления метода Гаусса, определитель матрицы системы (I.39).

Решение. Используя результаты предыдущего примера, нетрудно убедиться, что $\det A = 5 \cdot 3,2 \cdot 2,1875 = 35$.

Общий результат прямого и обратного хода описанной схемы метода Гаусса можно представить в матричной форме следующим образом.

Пусть U_k , $k = 1, 2, \dots, n$, матрицы порядка n вида

$$U_k = E + \xi^{(k)} e_k^T, \quad U_1 = E, \quad (I.40)$$

где $\xi^{(k)}$ есть n -мерный вектор с элементами

$$\xi_i^{(k)} = -a_{ik}^{(i)}, \quad i < k; \quad \xi_i^{(k)} = 0, \quad i \geq k \quad (I.41)$$

(см. рис. 2).

Нетрудно убедиться в справедливости равенства

$$U_1 \dots U_n U = E, \quad (I.42)$$

где U — матрица в разложении (I.35), (I.37).

Согласно (I.42)

$$U^{-1} = U_1 \dots U_n, \quad (I.43)$$

а следовательно, с учетом (I.33), (I.38), (I.43) искомое решение системы можно представить в виде

$$x = U_1 \dots U_n M_n M_{n-1} \dots M_1 b. \quad (I.44)$$

Для применения схемы единственного деления необходимо, чтобы все ведущие элементы $a_{ss}^{(s-1)}$, $s = 1, 2, \dots, n$, были отличны от нуля. Кроме того, близость ведущих элементов к нулю может привести к большой потере точности вычисленного решения. Для того чтобы устранить обрыв вычислительного процесса при делении на нуль или избежать потери точности при делении на величины, близкие к нулю, обычно реализуют схемы метода Гаусса, осуществляющие выбор наибольших элементов по строке, столбцу или по всей матрице. Например, в схеме метода Гаусса с выбором главного элемента по всей матрице порядок исключения неизвестных в заданной системе определяется следующим образом. Из элементов еще не приведенной подматрицы (активной матрицы) на s -м шаге, $s = 1, 2, \dots, n-1$, выбирается наибольший по модулю, называемый главным элементом s -го шага. Стоящее при нем неизвестное исключается по описанному выше правилу. Для удобства вычислений перед исключением этого неизвестного делают перестановку уравнений и неизвестных так, чтобы главный элемент занял левый верхний угол преобразуемой матрицы. Если на s -м шаге наибольший элемент выбирается только среди элементов s -го столбца (строки), то такая схема называется схемой с выбором главного элемента по столбцу (строке). Для решения систем линейных алгебраических уравнений на ЭВМ необходимо использовать метод Гаусса с тем или иным выбором главного элемента.

Решение системы линейных алгебраических уравнений методом Гаусса требует n операций деления, $\frac{1}{3}(n^3 + 3n^2 + n)$ операций умножения и $\frac{1}{6}(2n^3 + 3n^2 - 5n)$ операций сложения. Для реализации метода требуется хранить $n^2 + n + 1$ машинных слов.

В заключение кратко остановимся на решении систем с ленточными несимметричными матрицами, у которых $a_{ij} = 0$ при $i - j > m_1$ и $j - i > m_2$, где m_1 и m_2 — некоторые неотрицательные числа.

Схема метода Гаусса, аналогичная описанной выше и реализуемая с выбором главного элемента по столбцу, вполне пригодна для решения данных систем. Учет ленточного вида матрицы позволяет существенно уменьшить объем вычислений и требуемый объем памяти. Анализ упомянутой вычислительной схемы применительно к ленточной матрице показывает, что на каждом k -м шаге алгоритма, кроме последних

$m_1 - 1$ шагов, преобразуется $m_1 + 1$ строк матрицы. В результате исключаются m_1 ненулевых поддиагональных элементов k -го столбца. Экономия памяти в данном случае достигается за счет размещения элементов ленты исходной матрицы A в виде двухмерного массива G с размерами $n \times s$, где $s = \min(n, m_1 + m_2 + 1)$. На k -м шаге преобразования будут обрабатываться лишь те элементы G , которые расположены в строках от k -й до $(k + m_1)$ -й. Начиная с $(n + 1 - m_1)$ -го шага каждый раз обрабатывается на одну строку меньше.

Рассмотрим конкретный пример хранения элементов ленточной матрицы. Пусть матрица A имеет вид ($n = 6$, $m_1 = 2$, $m_2 = 1$)

$$A = \begin{bmatrix} a_{11} & a_{12} & & & & \\ a_{21} & a_{22} & a_{23} & & & \\ a_{31} & a_{32} & a_{33} & a_{34} & & \\ & a_{42} & a_{43} & a_{44} & a_{45} & \\ & 0 & a_{53} & a_{54} & a_{55} & a_{56} \\ & & & a_{64} & a_{65} & a_{66} \end{bmatrix}.$$

Тогда

$$G = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} & a_{45} \\ a_{53} & a_{54} & a_{55} & a_{56} \\ a_{64} & a_{65} & a_{66} & 0 \end{bmatrix},$$

где $s = (m_1 + m_2 + 1) = 4$.

Учет ленточной структуры целесообразно организовать лишь для случая $s \ll n$. Отметим, что в массиве G после выполнения всех шагов исключения можно хранить лишь элементы верхней треугольной матрицы U разложения $A = LU$ (см. [35, 38]). Информация о матрице L должна храниться в дополнительном массиве с размерами $n \times m_1$. Чтобы избавиться от этого массива при решении системы линейных алгебраических уравнений по упомянутой схеме Гаусса, необходимо одновременно с преобразованием матрицы A выполнять соответствующие преобразования над правыми частями системы.

3. Метод, основанный на LU -разложении (компактная схема метода Гаусса). Пусть теперь в нашем распоряжении имеется ЭВМ, которая позволяет аппаратно или программно (с небольшой потерей быстродействия) проводить вычисление скалярного произведения

$(\alpha, \beta) = \sum_{i=1}^n \alpha_i \beta_i$ с удвоенной точностью по методу накопления и округлением лишь окончательного результата. В этом случае для решения системы линейных алгебраических уравнений (I.7) с несимметричной плотной невырожденной матрицей, помещающейся в оперативную

память машины, целесообразно применять метод, основанный на непосредственном разложении $A = LU$. Этот метод сводится к непосредственному разложению матрицы A системы (I.7) в произведение нижней треугольной матрицы L и верхней треугольной матрицы U : $A = LU$:

$$Ax = LUx = b. \quad (I.45)$$

Это разложение, если оно существует, единственно с точностью до выбора диагональных элементов матриц L и U , а именно можно положить либо все $l_{ii} = 1$, либо все $u_{ii} = 1$. Рассмотрим подробно такой случай, когда U — есть верхняя треугольная матрица с единицами на главной диагонали. Формулы для вычисления элементов l_{ij} и u_{ij} легко получить, приравнявая соответствующие элементы в матричном равенстве

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \dots & u_{1n} \\ 0 & 1 & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \quad (I.46)$$

Такое разложение соответствует полученному в схеме единственного деления метода Гаусса.

Элементы матриц L и U на r -м шаге, $r = 1, 2, \dots, n$, вычислительного процесса определяются в следующем порядке: вначале элементы r -го столбца матрицы L

$$l_{ir} = a_{ir} - \sum_{k=1}^{r-1} l_{ik} u_{kr}, \quad i = r, \dots, n, \quad (I.47a)$$

а затем — r -й строки матрицы U

$$u_{ri} = \left(a_{ri} - \sum_{k=1}^{r-1} l_{rk} u_{ki} \right) \frac{1}{l_{rr}}, \quad i = r + 1, \dots, n. \quad (I.47b)$$

Рассмотрим теперь (I.45). Полагая $Ux = y$, получаем уравнение

$$Ly = b \quad (I.48)$$

с нижней треугольной матрицей, решение которого находим по формулам

$$y_i = \left(b_i - \sum_{k=1}^{i-1} l_{ik} y_k \right) \frac{1}{l_{ii}}, \quad i = 1, 2, \dots, n. \quad (I.49)$$

Неизвестное x находят из системы

$$Ux = y \quad (I.50)$$

с верхней треугольной матрицей, вычисляя

$$x_i = y_i - \sum_{k=i+1}^n u_{ik} x_k, \quad i = n, n-1, \dots, 1. \quad (I.51)$$

Пример 3. Методом LU -разложения решить систему линейных алгебраических уравнений (1.39).

Решение. По формулам (1.47) вычисляем элементы матриц L и U . В результате получаем

$$L = \begin{bmatrix} 5 & 0 & 0 \\ 4 & 3,2 & 0 \\ 1 & -2,2 & 2,1875 \end{bmatrix}, U = \begin{bmatrix} 1 & 0,2 & 0,4 \\ 0 & 1 & -0,1875 \\ 0 & 0 & 1 \end{bmatrix}.$$

Используя формулы (1.49), находим решение системы линейных алгебраических уравнений (1.48)

$$y_1 = 1,6; y_2 = -1,375; y_3 = 2,$$

а по формулам (1.51) — решение системы (1.50)

$$x_3 = 2, x_2 = -1, x_1 = 1.$$

Описанная схема LU -разложения может быть реализована только в случаях, когда ее элементы $l_{ii} \neq 0$. Кроме того, близость этих элементов к нулю может привести к большой потере точности вычисленного на машине решения. Чтобы избежать этого, решение системы линейных алгебраических уравнений методом LU -разложения нужно осуществлять с выбором наибольшего по модулю (главного) элемента, например в столбце. Для этого на каждом шаге r выполняются следующие операции. Вычисляются элементы l_{ir} (см. (1.47a)) и размещаются на местах a_{ir} , $i = r, \dots, n$. Если $\max_{i \geq r} |l_{ir}| = |l_{pr}|$, то строки r и p меняются местами, т. е. у всех элементов индексы строки p заменяют индексом r и наоборот. При этом запоминаются номера переставленных строк. Вычисляются элементы u_{ri} по формулам (1.47б) и размещаются на месте элементов a_{ri} , $i = r + 1, \dots, n$.

Для повышения точности вычислений целесообразно все скалярные произведения в формулах (1.47a), (1.47б), (1.49), (1.51) вычислять по методу накопления с округлением лишь окончательного результата. Применение LU -разложения на машине, не обеспечивающей вычисление скалярного произведения по методу накопления и округления лишь окончательного результата, никаких преимуществ перед методом исключения Гаусса не дает.

Решение системы линейных алгебраических уравнений методом LU -разложения требует n операций деления (если хранить значения $1/l_{ii}$, а не l_{ii}), $1/6 (2n^3 - 3n^2 + n)$ операций умножения, $1/6 (2n^3 - 3n^2 + n)$ операций сложения. Для реализации метода требуется $(n^2 + n + 1)$ машинных слов памяти, не считая программы.

Так как известно, что определитель произведения двух квадратных матриц равен произведению определителей перемножаемых матриц, то из равенства (1.46) сразу следует:

$$\det A = \det L \det U = (-1)^s \prod_{i=1}^n u_{ii}, \quad (1.52)$$

где s — число перестановок строк в процессе решения.

4. Метод, основанный на PU -разложении (метод отражения). Для решения систем линейных алгебраических уравнений (1.7) с несимметричными, невырожденными квадратными матрицами можно использовать

метод отражений, в котором не бывает обрыва процесса вследствие обращения в нуль элементов матрицы в ходе преобразования. Точность решения, получаемого методом отражений, будет намного выше, если скалярные произведения, имеющиеся в алгоритме, будут вычисляться с удвоенной точностью.

Метод отражений состоит в преобразовании исходной системы (1.7) порядка n к эквивалентной системе

$$A^{(n-1)} \mathbf{x} = \mathbf{b}^{(n-1)} \quad (1.53)$$

с верхней треугольной матрицей с помощью последовательности ортогональных матриц отражения P_r , $r = 1, 2, \dots, n-1$ (см. п. 8. параграфа 1.1).

На каждом r -м шаге процесса матрица и правая часть системы преобразуются по формулам

$$P_r A^{(r-1)} = A^{(r)}, \quad P_r \mathbf{b}^{(r-1)} = \mathbf{b}^{(r)}, \quad r = 1, 2, \dots, n-1, \quad (1.54)$$

$$A^{(0)} \equiv A, \quad \mathbf{b}^{(0)} \equiv \mathbf{b}.$$

Матрица отражения P_r в данном случае представляется в виде

$$P_r = E - k_r \mathbf{w}^{(r)} \mathbf{w}^{(r)T}, \quad (1.55)$$

где E — единичная матрица порядка n , k_r — некоторая константа, а n -мерный вектор $\mathbf{w}$ выбирается по заданному n -мерному вектору $\mathbf{z}$, являющемуся r -м столбцом матрицы $A^{(r-1)}$,

$$\mathbf{z}^T = [z_1, z_2, \dots, z_{r-1}, z_r, z_{r+1}, \dots, z_n] \equiv \\ \equiv [a_{1r}^{(1)}, a_{2r}^{(2)}, \dots, a_{r-1,r}^{(r-1)}, a_{r,r}^{(r-1)}, a_{r+1,r}^{(r-1)}, \dots, a_{nr}^{(r-1)}],$$

таким образом, чтобы преобразование вектора $P_r \mathbf{z}$ оставляло $r-1$ первых компонент вектора $\mathbf{z}$ без изменений и аннулировало (заменило нулями) последние $n-r$:

$$P_r \mathbf{z} = [z_1, z_2, \dots, z_{r-1}, \bar{z}_r, 0, \dots, 0]^T, \quad \bar{z}_r \equiv a_{rr}^{(r)}.$$

Нетрудно убедиться, что требуемые свойства матрицы отражения P_r будут обеспечены, если положить (см. п. 1 параграфа 1.1 и формулу (1.55))

$$\mathbf{w}^{(r)T} = [0, 0, \dots, 0, z_r, z_{r+1}, \dots, z_n] - \alpha_r \mathbf{1}_r^T, \quad (1.56)$$

где

$$\alpha_r = \left(\sum_{i=r}^n z_i^2 \right)^{1/2}, \quad (1.57)$$

$\mathbf{1}_r$ — r -й координатный вектор, $k_r = \frac{1}{\alpha_r^2 - \alpha_r z_r}$.

Для численной устойчивости знак α_r выбирается так, чтобы $\alpha_r^2 - \alpha_r z_r = \alpha_r^2 + |\alpha_r z_r|$. При этом получаем

$$P_r \mathbf{z} = [z_1, z_2, \dots, z_{r-1}, \alpha_r, 0, 0, \dots, 0]^T.$$

В рассматриваемом методе решения системы (1.7) в качестве вектора $\mathbf{z}$ на r -м шаге, $r = 1, 2, \dots, n-1$, берется r -й столбец матрицы

$A^{(r-1)}$, поэтому после $(n-1)$ -го шага получаем систему с верхней треугольной матрицей U

$$\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(n-1)} \\ b_2^{(n-1)} \\ \vdots \\ b_n^{(n-1)} \end{bmatrix},$$

где $U \equiv A^{(n-1)} = P_{n-1} \dots P_2 P_1 A$.

Учитывая ортогональность и симметричность матриц отражения, можно записать $A = P_1 P_2 \dots P_{n-1} U \equiv PU$. Таким образом, матрица A представляется в виде произведения ортогональной матрицы P и верхней треугольной U .

Решение системы с треугольной матрицей U осуществляется по формулам

$$x_i = \frac{b_i - \sum_{j=i+1}^n u_{ij} x_j}{u_{ii}}, \quad i = n, n-1, \dots, 1. \quad (I.58)$$

Необходимо отметить, что на практике не следует строить матрицу P_r , $r = 1, 2, \dots, n-1$, в явном виде, а выгодно оставить ее в факторизованной форме (I.55), так чтобы

$$P_r z = z - k C w^{(r)}, \quad \text{где } C = w^{(r)T} z \equiv (w^{(r)}, z).$$

Факторизованная форма хранения матрицы отражения особенно удобна, когда нужно решать систему с несколькими правыми частями, определяемыми после выполнения разложения $A = PU$. Всю необходимую для этого информацию можно сохранить, если для каждой матрицы P_r запоминать только $(n-r)$ ненулевых элементов соответствующего вектора $w^{(r)}$ (см. (I.56)). Эти элементы удобно разместить на местах, занимаемых последними $(n-r)$ элементами r -го столбца матрицы $A^{(r-1)}$. Согласно предыдущим рассуждениям все эти элементы вектора $w^{(r)}$ уже на месте, кроме r -го. Для восстановления r -го элемента вектора $w^{(r)}$ нужно отдельно хранить значения $\alpha_r \equiv u_{rr}$, $r = 1, 2, \dots, n-1$, т. е. диагональные элементы матрицы U . Значение коэффициента k_r на каждом шаге процесса может восстанавливаться по формуле (I.57).

Пример 4. Методом отражений решить систему линейных алгебраических уравнений

$$Ax = b, \quad (I.59)$$

где

$$A = \begin{bmatrix} 4 & 3 & 1 \\ 2 & -2 & 1 \\ 4 & 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 1 \\ 6 \end{bmatrix}. \quad (I.60)$$

Решение. Для $r = 1$ в качестве заданного вектора возьмем первый столбец матрицы $A = A^{(0)}$, т. е. $z^{(1)} = [4, 2, 4]^T$.

По столбцу $z^{(1)}$ строим вектор $w^{(1)}$ (I.56), а для этого вычисляем α_1 и k_1 по формуле (I.57):

$$\alpha_1^2 = 4^2 + 2^2 + 4^2 = 36,$$

$\alpha_1 = -6$, так как первая компонента z_1 вектора $z^{(1)}$ равна 4,

$$k_1 = \frac{1}{\alpha_1^2 - \alpha_1 z_1} = \frac{1}{36 + 24} = \frac{1}{60}.$$

Таким образом, вектор $w^{(1)T}$, вычисляемый по формуле (I.56), приобретает вид

$$w^{(1)T} = [10, 2, 4].$$

Зная значения k_1 и вектор $w^{(1)T}$, по формуле (I.55) легко построить матрицу

$$\begin{aligned} A^{(1)} = P_1 A = A - \frac{1}{60} \begin{bmatrix} 10 \\ 2 \\ 4 \end{bmatrix} w^{(1)T} A = A - \frac{1}{60} \begin{bmatrix} 10 \\ 2 \\ 4 \end{bmatrix} [60, 30, 16] = \\ = A - \begin{bmatrix} 10 \\ 2 \\ 4 \end{bmatrix} \begin{bmatrix} 1 & 1/2 & 4/15 \end{bmatrix}. \end{aligned}$$

После первого шага получаем

$$A^{(1)} \equiv P_1 A = \begin{bmatrix} -6 & -2 & -\frac{5}{3} \\ 0 & -3 & \frac{1,4}{3} \\ 0 & -1 & -\frac{0,2}{3} \end{bmatrix}, \quad b^{(1)} = P_1 b^{(0)} = \frac{1}{3} \begin{bmatrix} -29 \\ -7,6 \\ -3,2 \end{bmatrix}.$$

Для второго шага $r = 2$ имеем

$$z^{(2)} = [-2, -3, -1]^T, \quad \alpha_2^2 = \sum_{i=2}^3 z_i^2 = 10, \quad \alpha_2 = \sqrt{10},$$

$$k_2 = \frac{1}{\alpha_2^2 + |\alpha_2 z_2|} = \frac{1}{10 + 3\sqrt{10}}, \quad w^{(2)T} = [0, -3 - \sqrt{10}, -1],$$

$$A^{(2)} = P_2 A^{(1)} = \begin{bmatrix} -6 & -2 & -\frac{5}{3} \\ 0 & \sqrt{10} & -\frac{4}{3\sqrt{10}} \\ 0 & 0 & -\frac{2}{3\sqrt{10}} \end{bmatrix},$$

$$b^{(2)} = P_2 b^{(1)} = \frac{1}{3} \begin{bmatrix} -29 \\ \frac{26}{\sqrt{10}} \\ -\frac{2}{\sqrt{10}} \end{bmatrix}.$$

Таким образом, вместо (I.59), (I.60) получаем систему

$$\begin{aligned} -6x_1 - 2x_2 - \frac{5}{3}x_3 &= -\frac{29}{3}, \\ \sqrt{10}x_2 - \frac{4}{3\sqrt{10}}x_3 &= \frac{26}{3\sqrt{10}}, \\ -\frac{2}{3\sqrt{10}}x_3 &= -\frac{2}{3\sqrt{10}}. \end{aligned}$$

Решение полученной системы с треугольной матрицей (обратный ход) дает

$$x_3 = 1, \quad x_2 = 1, \quad x_1 = 1.$$

Определитель в этом случае находят по формуле

$$\det A = \det P \det U = (-1)^{n-1} \prod_{i=1}^n u_{ii},$$

поскольку определитель каждой матрицы отражения P_r , $r = 1, 2, \dots$, $n-1$, равен -1 .

Для решения системы линейных алгебраических уравнений порядка n с матрицей плотного заполнения методом, основанным на PU -разложении, необходимо выполнить $\frac{1}{6}(4n^3 + 15n^2 + 11n)$ умножений, $(2n-1)$ делений, $\frac{1}{6}(4n^3 + 9n^2 + 5n)$ сложений, $(n-1)$ извлечений квадратных корней. При этом требуется хранить $(n^2 + 2n + 1)$ элементов.

5. Метод, основанный на LL^T -разложении (метод квадратных корней). Для решения систем линейных алгебраических уравнений с симметричными положительно определенными матрицами целесообразно использовать метод, основанный на LL^T -разложении. Этот метод, реализуемый с учетом симметричности матрицы, оказывается самым экономным из прямых методов как по числу арифметических операций, так и по загрузке памяти электронной вычислительной машины. Если на ЭВМ возможна реализация вычислений скалярных произведений с накоплением, то метод, основанный на LL^T -разложении, позволяет получить решение системы линейных уравнений с более высокой точностью. Этот метод легко модифицируется на матрицы ленточной и профильной структуры и случай разреженных матриц, что позволяет существенно увеличить порядок решаемых систем и экономить время решения на ЭВМ.

Метод, основанный на LL^T -разложении, по существу является методом LU -разложения, примененным к системе линейных алгебраических уравнений (I.7) с симметричной матрицей. В этом методе симметричную матрицу A представляют в виде произведения двух взаимно транспонированных треугольных матриц

$$A = LL^T,$$

где

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}.$$

Формулы для вычисления элементов матрицы L имеют вид

$$\begin{aligned} l_{11} &= \sqrt{a_{11}}, \quad l_{i1} = a_{i1} \frac{1}{l_{11}}, \quad i = 2, 3, \dots, n, \\ l_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}, \quad l_{ij} = \frac{\left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk}\right)}{l_{ij}}, \\ i &= 2, 3, \dots, n, \quad j = 1, 2, \dots, i-1. \end{aligned} \quad (\text{I.61})$$

Таким образом, вместо системы (I.7) с симметричной матрицей рассматривают две системы уравнений с треугольными матрицами

$$Ly = b, \quad L^T x = y,$$

неизвестные в которых находят по следующим формулам:

$$\begin{aligned} y_1 &= b_1 \frac{1}{l_{11}}, \quad y_i = \left(b_i - \sum_{k=1}^{i-1} l_{ik} y_k\right) \frac{1}{l_{ii}}, \quad i = 2, 3, \dots, n, \\ x_n &= y_n \frac{1}{l_{nn}}, \quad x_i = \left(y_i - \sum_{k=i+1}^n l_{ki} x_k\right) \frac{1}{l_{ii}}, \\ i &= n-1, n-2, \dots, 1. \end{aligned} \quad (\text{I.62})$$

Метод, основанный на LL^T -разложении, для систем линейных алгебраических уравнений с симметричными и положительно определенными матрицами порядка n требует при реализации n делений, $\frac{1}{6}(n^3 + 9n^2 + 2n)$ умножений, $\frac{1}{6}(n^3 + 6n^2 - 7n)$ сложений и n извлечений квадратных корней. При реализации этого метода на электронной вычислительной машине необходимо хранить $\frac{1}{2}(n^2 + 3n + 2)$ элементов.

Из вычислительных формул метода следует, что для положительно определенной матрицы процесс разложения $A = LL^T$ всегда выполним и никаких перестановок не требуется. Заметим, что в случае симметричной, но не положительно определенной матрицы данное треугольное разложение матрицы A может не существовать без перестановок и в ходе вычислений появляются чисто мнимые числа, что усложняет вычислительную схему решения задачи. Поэтому в данном случае целесообразнее использовать метод, основанный на LU -разложении.

Метод, основанный на LL^T -разложении, удобно применять для решения системы уравнений с положительно определенной матрицей ленточного вида, т. е. такой, у которой $a_{ij} = 0$ при $|i - j| > m$, $m \ll n$.

Приведем пример матрицы ленточной структуры, у которой $n = 5$, $m = 1$:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{bmatrix}.$$

Из формул (I.61) очевидно, что при разложении ленточной матрицы получают $l_{ij} = 0$, если $i - j > m$, т. е. матрица L имеет тот же вид, что и нижняя половина ленточной матрицы A . Необходимо понимать, что речь идет о совпадении тех нулевых элементов матриц A и L , для которых $i \leq n$, $i - j > m$. Если же отдельные элементы a_{ij} или даже целые диагонали нижней половины матрицы A , расположенные внутри ленты ($|j - i| \leq m$), будут нулевыми, то отвечающие им элементы l_{ij} матрицы L , как правило, получаются отличными от нуля. Иными словами, полоса l_{ij} , $i \geq j$, $i - j \leq m$ будет сплошной.

Если ленточную положительно определенную матрицу A представить в виде произведения двух взаимно транспонированных матриц $A = LL^T$ и вычислять элементы l_{ij} матрицы L построчно, $i = 1, 2, \dots, n$, исключая обработку тех элементов, которые остаются нулевыми, то алгоритм данной модификации разложения матрицы A имеет вид для $i = 1, 2, \dots, n$, $j = r, r + 1, \dots, i - 1$,

$$l_{ij} = \frac{1}{l_{ii}} \left[a_{ij} - \sum_{k=r}^{i-1} l_{ik} l_{jk} \right], \quad r = \begin{cases} 1, & \text{если } i \leq m + 1, \\ i - m, & \text{иначе} \end{cases}$$

$$l_{ii} = \left[a_{ii} - \sum_{k=r}^{i-1} l_{ik}^2 \right]^{1/2},$$

Обратный ход осуществляется по следующим формулам:

$$y_i = \frac{1}{l_{ii}} \left[b_i - \sum_{k=r}^{i-1} l_{ik} y_k \right], \quad i = 1, 2, \dots, n,$$

$$x_i = \frac{1}{l_{ii}} \left[y_i - \sum_{k=i+1}^n l_{ki} x_k \right], \quad i = n, n - 1, \dots, 1,$$

$$t = \begin{cases} n, & \text{если } n - i \leq m, \\ i + m, & \text{иначе} \end{cases}$$

При решении систем линейных алгебраических уравнений с матрицей ленточного вида методом LL^T -разложения для хранения элементов a_{ij} нижнего треугольника исходной матрицы A можно использовать двухмерный массив G с размерами $(m + 1) \times n$, располагая в нем элементы a_{ij} , $i \geq j$, следующим образом:

$$G = \begin{bmatrix} a_{11} & a_{22} & a_{33} & \dots & a_{n-1,n-1} & a_{nn} \\ a_{21} & a_{32} & a_{43} & \dots & a_{n,n-1} & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m+1,1} & a_{m+2,2} & a_{m+3,3} & \dots & 0 & 0 \end{bmatrix},$$

т. е. столбцы G являются последовательными столбцами нижнего треугольника A .

Место расположения элемента a_{ij} в массиве G определяется по правилу

$$a_{ij} = g_{i-j+1,j} = g_{\alpha,\beta}, \quad 1 \leq \alpha \leq m + 1, \quad 1 \leq \beta \leq n,$$

т. е.

$$\alpha = i - j + 1, \quad \beta = j.$$

Модифицированные формулы (I.61), (I.62) можно переписать непосредственно относительно элементов массива G .

Для решения системы линейных алгебраических уравнений с положительно определенной матрицей ленточного вида (ширина ленты $2m + 1$) порядка n необходимо $0,5mn(m + 3) - m^3/3$ умножений, $m^2n/3$ сложений и n извлечений квадратного корня, а также память для хранения $(m + 2)n$ элементов матрицы и правой части.

Полезной модификацией метода, основанного на LL^T -разложении, является алгоритм, ориентированный на решение систем линейных алгебраических уравнений с положительно определенными профильными матрицами (определение профильной матрицы см. п. 8 параграфа I.1). Примером профильной матрицы может служить следующая:

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & a_{15} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & a_{42} & a_{25} & 0 & 0 & 0 & a_{29} \\ 0 & 0 & a_{33} & a_{34} & 0 & 0 & a_{37} & 0 & a_{39} \\ 0 & a_{42} & a_{43} & a_{44} & a_{45} & 0 & a_{47} & 0 & a_{49} \\ a_{51} & a_{52} & 0 & a_{54} & a_{55} & 0 & a_{57} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{66} & a_{67} & 0 & a_{69} \\ 0 & 0 & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & 0 & a_{79} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{88} & a_{89} \\ 0 & a_{92} & a_{93} & a_{94} & 0 & a_{96} & a_{97} & a_{98} & a_{99} \end{bmatrix}. \quad (I.63)$$

Как известно, все операции над симметричными положительно определенными матрицами можно выполнять, храня лишь элементы ее нижнего (или верхнего) треугольника в том числе главной диагонали. Поэтому для положительно определенной профильной матрицы $A = (a_{ij})$ будем называть профилем множество элементов a_{ij} , у которых $0 < i - j \leq m_i$, а размером профиля — число $q = \sum_{i=1}^n m_i$.

Для хранения информации о профильной матрице можно использовать различные схемы. Например, матрица n -го порядка профильного вида может храниться в одномерном массиве Q , где последовательно размещаются элементы строк нижнего треугольника исходной матрицы A , начиная с первого ненулевого. Для удобства выборки соответствующего элемента a_{ij} из массива $Q = \{Q(k)\}$ вводится дополнительный индексный одномерный массив длины n : $\mu = [\mu(1), \mu(2), \dots, \mu(n)]$, где $\mu(i)$ указывает на позицию диагональных элементов a_{ii} , $i = 1, 2, \dots$

..., n , в массиве Q . Тогда элемент a_{ij} , $i \geq j$, матрицы A находится в позиции k массива Q : $k = \mu(i) - i + j$. Заметим, что $\mu(n)$ указывает на длину массива Q , т. е. общее количество элементов в профиле и главной диагонали матрицы A . Для матрицы (1.63), $a_{ij} := a_{ji}$, получаем

$$Q = [a_{11}, a_{21}, a_{22}, a_{33}, a_{42}, a_{43}, a_{44}, a_{51}, a_{52}, 0, a_{54}, a_{55}, a_{66}, a_{73}, a_{74}, a_{75}, a_{76}, a_{77}, a_{88}, a_{92}, a_{93}, a_{94}, 0, a_{96}, a_{97}, a_{98}, a_{99}], \quad \mu = [1, 3, 4, 7, 12, 13, 18, 19, 27].$$

Модификация данной схемы хранения профильной матрицы предлагается в [28], где элементы профиля и диагональные элементы матрицы A хранятся в разных одномерных массивах: $Q = \{Q(k)\}$ длины q , $k = 1, 2, \dots, q$, и $D = \{D(i)\}$ длины n , $i = 1, 2, \dots, n$. Индексный массив $\mu = \{\mu(i)\}$ здесь имеет длину $n + 1$, $i = 1, 2, \dots, n + 1$, и строится по правилу:

$$\mu(i) = p(i-1) + 1, \quad i = 1, 2, \dots, n + 1,$$

где $p(i)$ — количество элементов профиля, содержащихся в первых i строках матрицы A , $p(0) = 0$, а $p(n) = q = \sum_{i=1}^n m_i$. Для отыскания в данном массиве Q позиции k элемента a_{ij} , $i > j$, принадлежащего профилю, достаточно использовать формулу

$$k = \mu(i+1) - i + j.$$

Для матрицы (1.63) при модифицированной схеме хранения имеем

$$Q = [a_{21}, a_{42}, a_{43}, a_{51}, a_{52}, 0, a_{54}, a_{73}, a_{74}, a_{75}, a_{76}, a_{92}, a_{93}, a_{94}, 0, a_{96}, a_{97}, a_{98}], \\ D = [a_{11}, a_{22}, a_{33}, a_{44}, a_{55}, a_{66}, a_{77}, a_{88}, a_{99}], \\ \mu = [1, 1, 2, 2, 4, 8, 8, 12, 12, 19].$$

Преимущество описанного варианта схемы хранения состоит в том, что он легко может быть модифицирован для случая несимметричной профильной матрицы, т. е. для матрицы с симметричным расположением ненулевых элементов a_{ij} , но при $a_{ij} \neq a_{ji}$.

В случае симметричной положительно определенной профильной матрицы A , хранимой по схеме [28], процесс разложения $A = LL^T$ удобно представить как вариант метода окаймления [28]. Для этого предположим, что матрица A порядка n представлена в виде

$$A = \begin{pmatrix} M_{n-1} & U \\ U^T & S \end{pmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1,n-1} & a_{1n} \\ a_{21} & \dots & a_{2,n-1} & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n-1,1} & \dots & a_{n-1,n-1} & a_{n-1n} \\ a_{n1} & \dots & a_{n,n-1} & a_{nn} \end{bmatrix},$$

и для положительно определенной подматрицы M_{n-1} порядка $(n-1)$ уже получено разложение $M_{n-1} = L_{n-1}L_{n-1}^T$, где L_{n-1} — нижняя

треугольная матрица порядка $(n-1)$:

$$L_{n-1} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \dots & \dots & \dots & \\ l_{n-1,1} & l_{n-1,2} & \dots & l_{n-1,n-1} \end{bmatrix}.$$

Тогда разложение $A = LL^T$ можно представить в виде

$$A = \begin{bmatrix} L_{n-1} & 0 \\ \mathbf{w}^T & t \end{bmatrix} \begin{bmatrix} L_{n-1}^T & \mathbf{w} \\ 0 & t \end{bmatrix},$$

где

$$L_{n-1}\mathbf{w} = \mathbf{U} \equiv \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{n-1n} \end{bmatrix},$$

$$\mathbf{w}^T = [l_{n1}, l_{n2}, \dots, l_{n,n-1}] \equiv (L_{n-1}^{-1}\mathbf{U})^T,$$

$$t = (S - \mathbf{w}^T\mathbf{w})^{1/2} = \left(a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2\right)^{1/2}.$$

Поскольку разложение подматрицы $M_{n-1} \equiv L_{n-1}L_{n-1}^T$ тоже можно получить посредством приема окаймления, то очевидно, что искомую матрицу L можно вычислять строка за строкой, начиная от подматрицы $M_1 = a_{11}$ порядка 1 и переходя к подматрицам M_i все большего порядка. Данная вычислительная схема описывается следующим образом: для вычисления элементов l_{ij} , $j = 1, 2, \dots, i$, i -й строки матрицы L нужно решить систему

$$L_{i-1}\mathbf{w} \equiv \begin{bmatrix} l_{11} & & 0 \\ \vdots & \ddots & \\ \vdots & & \\ l_{i-1,1} & \dots & l_{i-1,i-1} \end{bmatrix} \begin{bmatrix} l_{i,1} \\ \vdots \\ l_{i,i-1} \end{bmatrix} = \begin{bmatrix} a_{i1} \\ \vdots \\ a_{i,i-1} \end{bmatrix} \quad (1.64)$$

и вычислить $l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)^{1/2}$, $i = 1, 2, \dots, n$.

Применение изложенного способа разложения $A = LL^T$ профильной матрицы позволяет эффективно использовать то, что векторы $\mathbf{U}^T$, т. е. строки профильной матрицы, в основном короткие. Действительно, при решении системы $L_{i-1}\mathbf{w} = \mathbf{U}$, т. е. при вычислении i -й строки матрицы L , используется не вся матрица L_{i-1} , а только ее часть $\bar{L}_{i-1}$, соответствующая вектору $\mathbf{U}^T$, т. е. профильному отрезку i -й строки (рис. 3, a — A -плотная матрица, b — A -профильная матрица).

После получения разложения $A = LL^T$ для вычисления решения системы $A\mathbf{x} = \mathbf{b}$ достаточно, как обычно, решить две треугольные системы

$$L\mathbf{y} = \mathbf{b}, \quad L^T\mathbf{x} = \mathbf{b}.$$

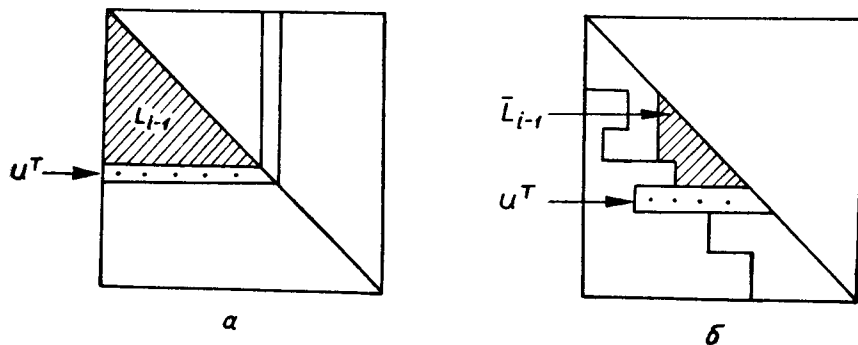


Рис. 3

Отметим, что в процессе разложения $A = LL^T$ для построения i -й строки матрицы L используется та же подпрограмма, по которой решается система $Ly = b$.

С деталями организации вычислений при программной реализации описанного алгоритма можно ознакомиться по работе [28].

6. Некоторые характеристики прямых методов решения и рекомендации по их использованию. Нами рассмотрены численные методы решения систем линейных алгебраических уравнений с несимметричными неособенными матрицами: метод Гаусса, методы, основанные на LU - и PU -разложениях, и метод квадратного корня для решения систем с симметричными и положительно определенными матрицами. Каждый из рассмотренных методов характеризуется своим числом арифметических операций, необходимых для реализации алгоритма на ЭВМ, и требованиями к памяти машины. Кроме того, представляет определенный интерес сравнение рассматриваемых методов по точности. Если использовать понятие эквивалентного возмущения исходных данных для определения суммарного влияния ошибок округления (см. п. 5 параграфа 1.2), то по оценкам норм этого возмущения можно получить некоторое представление о точности рассматриваемых методов.

Так как разложение матрицы A заданной системы (1.7) в произведение двух треугольных или ортогональной и треугольной матриц является главной частью описанных ранее методов², то целесообразно сравнить точность рассматриваемых методов по значению этой главной части эквивалентного возмущения. Обозначим реально вычисленные матрицы разложений через Q и P , тогда

$$QP = A + F,$$

где F — соответствующее эквивалентное возмущение. В табл. 1 для рассмотренных нами прямых методов приведены величины, характеризующие оценки норм F , полученных в работах [13, 15, 97]:

$$\|F\|_E \leq 2^{-t} f(n) \|A\|_E, \quad (M \|F\|_E)^{1/2} \leq 2^{-t} \Phi(n) (M \|A\|_E)^{1/2}.$$

² Отметим, что в процессе обратного хода происходит очень малое накопление ошибок округления.

В приведенных оценках эквивалентного возмущения предполагается, что на ЭВМ используется двоичная система счисления в режиме плавающей запятой с t разрядной мантисой у стандартного машинного представления числа, а накопление скалярного произведения происходит с $2t$ -разрядной мантисой и округлением окончательной суммы до t двоичных цифр.

В табл. 1 c_i, k_i — константы, не зависящие от $n, g(n)$ — величина, зависящая от способа выбора главного элемента и характеризующая рост элементов матрицы в процессе преобразования. Отметим, что при выборе главного элемента по всей матрице

$$g(n) < 1,8n^{\frac{1}{4} \lg n} \max_{i,j} |a_{ij}|$$

(см. [110]). Есть предположение, что для всех действительных матриц

$$g(n) \leq n \max_{i,j} |a_{ij}|.$$

При выборе главного элемента по столбцу

$$g(n) \leq 2^{n-1} \max_{i,j} |a_{ij}|.$$

Анализ суммарных ошибок округления прямых методов показал, что с точки зрения устойчивости к ошибкам округления между рассмотренными методами принципиальной разницы нет. Погрешность машинной реализации определяется, с одной стороны, числом обусловленности матрицы, ее порядком, с другой — методом решения, способом выбора главного элемента и длиной мантиссы. Чем большее значение t , тем меньшие эквивалентные возмущения и тем меньшие ошибки машинной реализации.

Для решения систем линейных алгебраических уравнений с положительно определенными матрицами (плотными, ленточного вида) наилучшим (по использованию памяти, по быстродействию и точности получаемого решения, если возможна реализация скалярного произведения с накоплением и округлением окончательного результата) является метод, основанный на LL^T -разложении. Для решения систем линейных алгебраических уравнений с невырожденными матрицами произвольного вида или невырожденными симметричными знакоопределенными матрицами при возможности вычисления на машине скалярного произведения по методу накопления целесообразно применять метод, основанный на LU -разложении, с тем или иным выбором главного элемента. Применение процедуры вычисления скалярного произведения по методу накопления существенно повышает точность расчетов.

Таблица 1

Метод	$f(n)$	$\Phi(n)$
Гаусса с выбором главного элемента	$c_1 g(n) n$	$k_1 g(n) \sqrt{n}$
Краута с выбором главного элемента	$c_2 g(n)$	—
Квадратного корня для положительно определенной матрицы	c_3	$k_2 \frac{1}{\sqrt{n}}$
Отражений	$c_4 n$	$k_3 \sqrt{n}$

Если на ЭВМ отсутствуют аппаратные средства вычисления скалярного произведения по методу накопления, а программная их реализация существенно замедляет быстродействие машины, то для решения систем с невырожденной произвольной матрицей целесообразно использовать метод Гаусса с выбором главного элемента. Практически всегда применяют частичный выбор, т. е. выбор главного элемента по столбцу или строке.

Заметим, что если матрица системы невырождена (выполняется условие (I.9)), то увеличением длины машинного слова ЭВМ всегда можно добиться близости машинного к математическому решению задачи.

Прямые методы допускают различные модификации, позволяющие использовать память второй ступени. Иногда в оперативную память из памяти второй ступени вводится матрица последовательно по строкам. В некоторых случаях матрицу большого порядка разбивают на блоки и организуют решение систем прямыми методами с учетом блочности матрицы.

1.4. ОЦЕНКИ ДОСТОВЕРНОСТИ РЕШЕНИЙ, ПОЛУЧАЕМЫХ ПРЯМЫМИ МЕТОДАМИ

1. Процедура итерационного уточнения решения. В параграфе I.2 рассматривались теоретические аспекты близости математического к физическому и машинного к математическому решениям задач. Рассмотрим теперь некоторые практические приемы, позволяющие оценивать достоверность полученного на ЭВМ решения.

Представляет большой интерес вопрос о близости машинного решения $x^{(1)}$ к математическому решению системы (I.7). Действительно, машинное решение есть точное решение возмущенной системы (I.20). Для упрощения будем считать, что погрешность перевода исходных данных из десятичной системы счисления в двоичную равна нулю (исходные данные заданы точно). Рассмотрение невязки $z = b - Ax^{(1)}$ для оценки точности вычисленного решения иногда оказывается бесполезным, так как далекие от точного решения векторы могут давать очень малые невязки. Это случается в системах, у которых норма обратной матрицы велика. Действительно, если точное решение системы (I.7) представить в виде $x = x^{(1)} + \delta$, то для вычисления поправки получаем систему

$$A\delta = b - Ax^{(1)} \equiv r^{(1)}.$$

Отсюда следует, что $\delta = A^{-1}r^{(1)}$ и

$$\|\delta\| \leq \|A^{-1}\| \|r^{(1)}\|.$$

Очевидно, даже при малой невязке погрешность δ в решении может оказаться весьма большой, если норма $\|A^{-1}\|$ велика.

Получить информацию о близости машинного решения $x^{(1)}$ к математическому решению x , а в ряде случаев устранить погрешность машинной реализации можно с помощью итерационного процесса [см.

например, работы 97, 98]:

$$r^{(s)} = b - Ax^{(s)}, \quad QP\delta^{(s)} = r^{(s)}, \quad x^{(s+1)} = x^{(s)} + \delta^{(s)}, \quad (I.65)$$

$$s = 0, 1, 2, \dots, \quad x^{(0)} \equiv 0, \quad \delta^{(0)} \equiv x^{(1)}.$$

Поправки $\delta^{(s)}$ ($s = 1, 2, \dots$) находят, используя уже полученное при вычислении $x^{(1)}$ реальное разложение матрицы A в произведение двух матриц: $A + F = QP$, т. е. вычисление $\delta^{(s)}$ ($s \geq 1$) сводится к соответствующей обработке правых частей и вычислениям по формулам обратного хода. Поэтому такое итерационное уточнение решения требует небольшого дополнительного времени по сравнению с первоначальным решением системы и может быть организовано как решение выбранным прямым методом одной системы с разными последовательно вводимыми правыми частями. В формулах (I.65) невязки $r^{(s)}$ необходимо вычислять с повышенной точностью за счет либо накопления скалярного произведения, либо удвоения длины машинного слова на этом этапе вычислений по сравнению с длиной машинного слова в основном расчете. В случае не слишком «плохой» машинной обусловленности матрицы системы последовательность сходится к точному (в пределах длины машинного слова) решению системы (I.19) [97, 98]. Применять процедуру (I.65) есть смысл только при вычислении невязок с повышенной точностью.

Практически матрица A рассматривается как «плохо машинно обусловленная», если наблюдается неубывание норм двух последовательных поправок $\delta^{(s)}$ или чрезмерно медленное их убывание: за итерацию уточняется менее двух двоичных или половины десятичного разряда. Заметим, что термин «плохая машинная обусловленность» матрицы относится к данной конкретной длине машинного слова.

Проиллюстрируем применение процесса (I.65) на примере решения системы (I.17) [77]. Для решения системы на ЭВМ МИР-1 выбрана длина машинного слова шесть десятичных знаков (см. п. 5 параграфа I.2). В ходе итерационного процесса (I.65) получены следующие результаты:

$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
-0,037 884 8	-0,514 729	-1,302 137	-3,156 56
0,764 402	1,052 429	1,527 90	2,647 87
0,710 031	0,829 836	1,027 89	1,493 59
0,434 782	0,363 669	0,246 25	-0,030 055.

Невязки в этом случае имели вид

$r^{(1)}$	$r^{(2)}$	$r^{(3)}$
-0,245 · 10 ⁻⁶	0,5 · 10 ⁻⁸	0,346 · 10 ⁻⁵
0,350 6 · 10 ⁻⁶	-0,129 · 10 ⁻⁶	0,463 6 · 10 ⁻⁵
-0,125 62 · 10 ⁻⁵	-0,316 · 10 ⁻⁶	0,418 8 · 10 ⁻⁵
-0,145 56 · 10 ⁻⁵	-0,172 · 10 ⁻⁶	0,429 7 · 10 ⁻⁵

По существу невязки были равны машинному нулю, хотя на самом деле решение не было получено. На этом примере видна несостоятельность критерия невязок для оценки точности машинного решения.

Поправки, вычисленные в ходе итерационного процесса:

$\delta^{(1)}$	$\delta^{(2)}$	$\delta^{(3)}$
0,476 845	-0,787 408	-1,854 371
0,288 028	0,475 488	1,119 974
-0,119 805	0,197 993	0,465 775
-0,071 113	-0,117 465	-0,276 260

свидетельствуют о его расходимости. Учитывая поведение поправок, можно утверждать, что несмотря на малость невязок для машинного слова в шесть десятичных знаков, нами получено машинное решение задачи, весьма далекое от математического.

Для приближения машинного решения к математическому необходимо увеличить длину машинного слова. Для машинного слова в десять десятичных знаков на МИР-1 были получены следующие результаты:

$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
0,399 534 697 9	0,399 995 465 8	0,400 000 074 2	0,4
0,500 281 036 7	0,500 002 736 8	0,499 999 955 2	0,5
0,600 116 836 2	0,600 001 146 9	0,599 999 981 3	0,6
0,499 930 707 8	0,499 999 323 0	0,500 000 011 1	0,5

$\delta^{(1)}$	$\delta^{(2)}$	$\delta^{(3)}$
0,460 767 908 $\cdot 10^{-3}$	0,460 845 392 $\cdot 10^{-5}$	-0,742 007 060 $\cdot 10^{-7}$
0,278 298 016 $\cdot 10^{-3}$	-0,278 162 669 $\cdot 10^{-5}$	0,448 004 255 $\cdot 10^{-7}$
0,115 691 252 $\cdot 10^{-3}$	-0,115 961 132 $\cdot 10^{-5}$	0,187 001 849 $\cdot 10^{-7}$
0,686 151 867 $\cdot 10^{-4}$	0,688 106 749 $\cdot 10^{-6}$	-0,111 001 075 $\cdot 10^{-7}$

Четвертая итерация дает точное решение исходной системы. Даже если нам не известно точное решение, то поведение последовательных поправок позволяет сделать вывод о сходимости итерационного процесса.

Таким образом, процедура (I.65) позволяет устранить погрешность реализации алгоритма и получить решение системы (I.19).

2. Апостериорные оценки числа обусловленности системы. Сходящийся итерационный процесс (I.65) можно использовать для получения информации о числе обусловленности матрицы системы линейных алгебраических уравнений (I.7). Зная погрешность задания исходных данных и приближенное число обусловленности матриц, по формуле (I.16) можно вычислить число обусловленности системы. Иными словами, появляется возможность оценить близость математического и физического решений задачи.

Поведение последовательных поправок $\delta^{(s)}$, $s = 1, 2, \dots$, при их заметном убывании позволяет определить число верных десятичных цифр полученного решения

$$\eta = \left\lfloor -\lg \frac{\|\delta^{(1)}\|_{\infty}}{\|x^{(1)}\|_{\infty}} \right\rfloor.$$

Справедлива следующая приближенная оценка числа обусловленности матрицы системы (I.7)

$$\text{cond } A \approx \frac{1}{\varepsilon} \frac{\|\delta^{(1)}\|}{\|x^{(1)}\|} \approx \frac{1}{\varepsilon} 10^{-\eta}, \quad (1.66)$$

где ε — наибольшее число, для которого равенство $1,0 + \varepsilon = 1,0$ справедливо в вычислениях на ЭВМ с плавающей запятой [109].

При решении на вычислительной машине МИР-1 системы (I.17) оказалось, что итерационный процесс уточнения решения сходящийся, и нами было получено машинное решение, совпадающее с математическим решением задачи. В этих условиях, используя формулу (I.66), можно получить приближенное число обусловленности матрицы в системе (I.17):

$$\text{cond } A \approx 10^{10} \frac{\|\delta^{(1)}\|_{\infty}}{\|x^{(1)}\|_{\infty}} \approx 0,77 \cdot 10^7,$$

которое хорошо согласуется с истинным

$$\text{cond } A = \|A\|_{\infty} \|A^{-1}\|_{\infty} = 1,34 \cdot 10^7.$$

Используя приближенное значение $\text{cond } A$, можно вычислить число обусловленности m системы (I.7) по формуле (I.16) и оценить по формулам (I.14), (I.15) относительную погрешность математического решения задачи. Если окажется, что относительная погрешность велика, то к такому математическому результату следует подходить с осторожностью. В этом случае полезно повысить точность задания исходных данных и, если это невозможно, задачу следует переформулировать относительно других параметров.

Таким образом, итерационный процесс (I.65) позволяет не только оценивать близость машинного решения к математическому, но и вычислять близость математического решения к искомому решению физической модели.

Отметим, что приближенную оценку числа обусловленности матрицы системы можно получить, используя лишь ранее найденное разложение $A = QP$, а не итерационный процесс (I.65). Этот способ сводится к решению систем

$$(QP)^T z = f, \quad QPy = z$$

со специально подобранным вектором f . Он содержит только $O(n^2)$ арифметических операций, где n — порядок матрицы A . Преимущество его состоит в том, что не требуется организации процедуры итерационного уточнения. Подробное описание алгоритма оценки дано в работе [140] (см. также [110]).

При решении на ЭВМ систем линейных алгебраических уравнений, описывающих прикладные задачи, не всегда машинное решение будет достаточно близко к математическому, а математическое — к решению физической модели. Вместо системы с точными исходными данными возникает алгебраическая задача, в которую входит сама система и

соотношения, характеризующие точность задания коэффициентов и правой части.

При рассмотрении такой задачи сначала надо определить, что будем подразумевать под решением задачи, а в зависимости от этого и строить методы нахождения определенного математического решения.

Зная физику процесса и используемый математический аппарат его описания, априори можно определить, обладает рассматриваемая математическая модель единственным решением или этих решений множество. Для решения систем линейных алгебраических уравнений, описывающих прикладные задачи, которые обладают неединственным решением, с прямоугольными или квадратными матрицами разыскивают одно из обобщенных решений, которое находят специальными прямыми и итерационными методами, рассмотренными в гл. 4 (см. также [91, 94, 98, 105]).

Если рассматриваемая задача обладает единственным решением, то необходимо в ходе вычислений установить обусловленность системы линейных алгебраических уравнений. Решение плохо обусловленной системы, являясь неустойчивым относительно малых изменений входных данных, имеет устойчивую проекцию на подпространство, образованное собственными векторами матрицы $A^T A$, отвечающими «большим» собственным значениям [103]. Если нельзя переформулировать прикладную задачу так, чтобы получить достаточно хорошо обусловленную систему (например, путем перехода к исследованию других параметров физического процесса), то для решения плохо обусловленной системы линейных алгебраических уравнений строят лишь упомянутую устойчивую проекцию [16, 100, 102]. Как отмечается в работе [15], для решения плохо обусловленных задач невозможно построить достаточно эффективное решение без привлечения в процессе вычислений дополнительной априорной информации. В целом вопрос о решении плохо обусловленных систем линейных алгебраических уравнений далек от завершения.

Для решения хорошо обусловленных систем линейных алгебраических уравнений в зависимости от вида и свойств матрицы системы, а также характеристик и свойств имеющихся вычислительных машин, существуют различные устойчивые вычислительные схемы прямых методов решения, которые в сочетании с процедурой итерационного уточнения позволяют успешно решать большинство возникающих на практике задач, оценивать близость машинного к математическому решению, уменьшать погрешность машинной реализации алгоритмов и давать в ряде случаев оценку наследственной погрешности получаемых решений.

Иногда перед применением прямых методов целесообразно провести масштабирование системы линейных алгебраических уравнений (иными словами, умножение матрицы слева и справа на диагональные матрицы D_1 и D_2 и соответствующее изменение правой части) с целью уменьшения числа обусловленности новой преобразованной матрицы. Система уравнений

$$Ax = b$$

с матрицей A преобразуется при этом в систему

$$D_1 A D_2 y = D_1 b, \quad x = D_2 y.$$

Удачное масштабирование может изменить к лучшему течение вычислительного процесса, повышая машинную устойчивость вычисляемого решения к влиянию ошибок округления. Наряду с двухсторонним масштабированием для некоторых целей полезным оказывается переход от матрицы A к матрице AD или DA . (О масштабировании см., например, в [104, 125, 126] и др.)

Для решения систем линейных алгебраических уравнений с симметричными положительно определенными матрицами различных видов целесообразно применять метод, основанный на LL^T -разложении.

При решении систем линейных алгебраических уравнений с невырожденными матрицами общего вида можно использовать метод, основанный на LU -разложении, т. е. описанный выше метод исключения Гаусса или его компактную схему с выбором главного элемента.

В последнее время начали эффективно применяться прямые методы, ориентированные на решение систем с симметричными не знакоопределенными матрицами. В основе этих методов лежат различные формы разложения исходной матрицы A на множители, где требуется порядка $\frac{1}{6} n^3$ умножений и $\frac{1}{6} n^3$ сложений. Назовем лишь две из них: разложение Аасена [122] и разложение Банча [135], которые существуют для любой симметричной матрицы и устойчивы. В первом случае разложение

имеет вид $A = \tilde{U} T \tilde{U}^T$, где $\tilde{U}$ — произведение элементарных верхних треугольных матриц и матриц перестановок, T — симметричная трехдиагональная матрица; во втором случае $A = V D V^T$, где V — произведение элементарных верхних треугольных матриц и матриц перестановок, D — симметричная блочно-диагональная матрица с блоками порядка 1 или 2.

Во всех методах, основанных на прямом разложении матрицы A , скалярные произведения целесообразно вычислять с удвоенной точностью.

В настоящее время разрабатываются прямые методы решения систем линейных алгебраических уравнений с разреженными матрицами, т. е. матрицами, у которых количество ненулевых элементов порядка n (см., например, [95]). Возможности применения прямых методов к системам линейных алгебраических уравнений с разреженными матрицами разнообразны. Отметим некоторые из них. При решении ряда дифференциальных задач методом конечных разностей или конечных элементов могут возникать матрицы, обладающие определенной спецификой (ленточные, профильные, допускающие разложение на «редкие» клетки и т. д.). Для этих методов могут быть выписаны модификации известных прямых методов или созданы новые методы, специфика которых определяется дифференциальной (разностной) задачей (например, метод матричной декомпозиции [136]).

Иногда «удобная» форма матрицы обнаруживается после надлежащего изменения порядка следования строк и столбцов. В некоторых работах (см., например, [28]) исследуется вопрос о приведении матрицы

к блочно-треугольному или блочно-диагональному виду. Разработаны специальные методы приведения матриц к ленточным матрицам, имеющих сравнительно небольшую ширину ленты (см., например, [28, 139]).

Иногда сравнительно небольшое число элементов разреженной матрицы препятствует приведению к одной из удобных форм. Если их заменить нулями, то для возвращения к исходной матрице нужно добавить поправочную матрицу, которая имеет очень мало ненулевых элементов и соответственно малый ранг. Для решения таких задач целесообразно применять методы, рассмотренные в работе [193].

Для решения систем с разреженными матрицами созданы специальные методы упаковки этих матриц и модификации прямых методов (Гаусса, основанного на LU -, LL^T -разложениях, Гаусса — Жордана и др.), обеспечивающие такой выбор порядка действий, при котором главный элемент был бы не меньше некоторого допустимого значения и при котором локальное заполнение было бы наименьшим [95].

После применения прямых методов решения целесообразно использовать процедуру итерационного уточнения решения. Если поправки в итерационной процедуре убывают сравнительно быстро, то последовательность векторов $x^{(s)}$ позволяет получить машинное решение, совпадающее в пределах длины машинного слова с математическим решением задачи.

После того как получено машинное решение, совпадающее с математическим решением задачи, необходимо оценить числа обусловленности матрицы и системы, характеризующие близость полученного математического решения задачи к физическому.

Иногда применяют как априорные, так и апостериорные оценки, характеризующие отклонение полученного машинного решения от решения задачи с учетом погрешностей входных данных и ошибок округления. Такие оценки зависят от задачи и от метода ее решения.

В так называемой интервальной арифметике вместо вещественных чисел задаются содержащие их интервалы и даются правила арифметических действий. В ответе тоже получают вектор-интервал, в котором содержатся компоненты решения задачи (см., например, [161]).

В связи с распространением мультимикропроцессорных машин начинают развиваться прямые методы решения систем в режиме параллельных вычислений (см., например, [159] и др.). Большое значение приобретают в настоящее время работы по организации библиотек и пакетов программ [69, 98].

Учитывая изложенное выше, отметим, что в настоящее время исследования в области решения систем линейных алгебраических уравнений прямыми методами сосредотачиваются вокруг следующих вопросов: решения систем с прямоугольными плохо обусловленными матрицами; построения решения больших систем линейных алгебраических уравнений с разреженными матрицами различной формы; создания надежных методов оценок достоверности получаемых на ЭВМ решений с использованием как априорной, так и апостериорной информации о решениях; организации параллельных вычислений на мультимикропроцессорных машинах; создания комплексов программных средств, реализующих алгоритмы прямых методов на ЭВМ.

ГЛАВА II

ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

II.1. НЕКОТОРЫЕ СВЕДЕНИЯ ИЗ ТЕОРИИ ЛИНЕЙНЫХ ВЕКТОРНЫХ ПРОСТРАНСТВ

1. Линейное векторное пространство. Введем его понятие. Пусть имеем совокупность произвольных элементов x, y, z , для которых определены две операции — сложение и умножение на число из поля K^3 , причем эти действия удовлетворяют следующим аксиомам:

- 1) $x + y = y + x$,
- 2) $(x + y) + z = x + (y + z)$,
- 3) существует такой элемент Θ , что $0x = \Theta$,
- 4) $1 \cdot x = x$,
- 5) $\alpha(\beta x) = (\alpha\beta)x$,
- 6) $(\alpha + \beta)x = \alpha x + \beta x$,
- 7) $\alpha(x + y) = \alpha x + \alpha y$,

где α, β — числа из K .

Любая совокупность элементов R , для которой всегда выполнимы и однозначны операции сложения и умножения элементов на число из K , удовлетворяющие аксиомам 1—7, называется линейным векторным пространством. Элементы векторного пространства R называются векторами. Пространство называется действительным, если для его векторов определено умножение на действительные числа, и — комплексным, если для его векторов определено умножение на комплексные числа.

Важным примером линейного векторного пространства является арифметическое (численное) пространство. Векторами этого пространства являются упорядоченные совокупности из n чисел, которые называются компонентами вектора. Сложение и умножение вектора на число определяются покомпонентно, т. е. суммой векторов $x = (x_1, x_2, \dots, x_n)$ и $y = (y_1, y_2, \dots, y_n)$ будет вектор

$$x + y = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n),$$

³ Под числовым полем понимают любую совокупность чисел, в пределах которой всегда выполнимы и однозначны четыре операции: сложение, вычитание, умножение и деление на число, отличное от нуля.

произведение числа α из K на x — вектор

$$\alpha x = (\alpha x_1, \alpha x_2, \dots, \alpha x_n).$$

Нулевым вектором будет вектор, все компоненты которого равны нулю. Легко проверить, что аксиомы 1—7 выполняются.

2. Линейная зависимость векторов. Линейной комбинацией векторов $x_1, x_2, \dots, x_n$ называется вектор $y = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$, где $c_1, \dots, c_n$ — числа из K . Пространство называется конечномерным, если существует конечное число векторов $x_1, x_2, \dots, x_n$ таких, что любой вектор пространства представляется в виде

$$c_1 x_1 + c_2 x_2 + \dots + c_n x_n.$$

Размерностью конечномерного пространства называется наименьшее число векторов, удовлетворяющих последнему требованию. Обозначим через R_n n -мерное векторное пространство.

Векторы $x_1, x_2, \dots, x_n$ пространства R_n называются линейно независимыми, если существуют числа $c_1, c_2, \dots, c_n$ из K , не все равные нулю и такие, что

$$c_1 x_1 + c_2 x_2 + \dots + c_n x_n = 0. \quad (\text{II.1})$$

Данные векторы линейно зависимы тогда и только тогда, когда один из них является линейной комбинацией остальных.

Если равенство (II.1) возможно лишь в том случае, когда $c_1 = c_2 = \dots = c_n$, то векторы $x_1, x_2, \dots, x_n$ называются линейно независимыми.

3. Базис n -мерного линейного пространства. Любая совокупность n линейно независимых векторов пространства R_n называется базисом этого пространства.

Каждый вектор пространства R_n может быть представлен и при этом единственным образом в виде линейной комбинации векторов базиса, т. е.

$$x = c_1 e_1 + c_2 e_2 + \dots + c_n e_n, \quad (\text{II.2})$$

где $e_1, e_2, \dots, e_n$ — базис пространства R_n . Числа $c_1, c_2, \dots, c_n$ называются координатами вектора x в данном базисе. Заметим, что компоненты вектора арифметического пространства

$$x = (x_1, x_2, \dots, x_n)$$

есть координаты его в базисе

$$e_j = (\delta_{1j}, \delta_{2j}, \dots, \delta_{nj}), \quad j = 1, 2, \dots, n,$$

где δ_{ij} — символ Кронекера. Следовательно, имеем разложение

$$x = x_1 e_1 + x_2 e_2 + \dots + x_n e_n. \quad (\text{II.3})$$

4. Линейное подпространство. Совокупность R_k векторов из n -мерного пространства R_n называется линейным подпространством R_n , если выполнены условия: из $x \in R_k$ и $y \in R_k$ следует $x + y \in R_k$, из $x \in R_k$ следует $\alpha x \in R_k$, где α — любое число из K .

Следовательно, R_k можно также считать векторным пространством. Максимальное число k линейно независимых векторов в пространстве R_k называется размерностью этого подпространства.

Если $z_1, z_2, \dots, z_m$ — векторы пространства R_n , то совокупность всех векторов вида

$$x = \alpha_1 z_1 + \alpha_2 z_2 + \dots + \alpha_m z_m, \quad (\text{II.4})$$

где α_j ($j = 1, 2, \dots, m$) — произвольные числа из K , представляет собой подпространство пространства R_n , причем если векторы $z_1, z_2, \dots, z_k$, $k \leq n$ линейно независимы, то размерность этого подпространства k . И наоборот, всякое подпространство R_k пространства R_n совпадает с совокупностью всех линейных комбинаций базисных векторов $z_1, z_2, \dots, z_k$ этого подпространства.

Совокупность векторов x , определяемых формулой (II.4), представляет собой наименьшее линейное пространство, содержащее векторы $z_1, z_2, \dots, z_k$ (так называемое пространство, порожденное векторами $z_1, z_2, \dots, z_k$, или пространство, натянутое на векторы $z_1, z_2, \dots, z_k$).

5. Скалярное произведение векторов. В комплексном (вещественном) пространстве R можно ввести понятие скалярного произведения следующим образом. Каждым двум векторам x и y из R сопоставляется некоторое комплексное (вещественное) число, обозначаемое (x, y) , причем для комплексного пространства должны быть удовлетворены для любых векторов x, y, z из R и любого комплексного числа α следующие аксиомы⁴:

- 1) $(x, y) = \overline{(y, x)}$,
- 2) $(x, x) > 0$, если $x \neq 0$ и $(x, x) = 0$, если $x = 0$,
- 3) $(x + y, z) = (x, z) + (y, z)$,
- 4) $(\alpha x, y) = \alpha (x, y)$.

Для вещественного пространства α — вещественное число, и аксиома 1 имеет вид $(x, y) = (y, x)$.

Вещественное линейное пространство с определенным таким образом скалярным произведением называется евклидовым пространством. Евклидово n -мерное пространство будем обозначать так: E_n .

С помощью скалярного произведения можно дать определение основных метрических понятий в векторном пространстве: длины вектора и угла между парой векторов. Длиной вектора пространства R называют число $|x| = \sqrt{(x, x)}$. Скалярное произведение двух векторов удовлетворяет неравенству

$$|(x, y)| \leq |x| |y|,$$

которое называется неравенством Коши — Буняковского. В случае евклидова пространства угол φ между ненулевыми векторами x и y определяется из формулы

$$\cos \varphi = \frac{(x, y)}{|x| |y|}.$$

На основании неравенства Коши — Буняковского такое определение угла всегда справедливо.

⁴ Черта над числом означает переход к комплексно-сопряженному числу.

Для векторов арифметического комплексного или вещественного пространства

$$\mathbf{x} = (x_1, x_2, \dots, x_n), \mathbf{y} = (y_1, y_2, \dots, y_n),$$

скалярное произведение вводится соответственно по формуле

$$(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n x_i \bar{y}_i, \quad (\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n x_i y_i.$$

Легко проверить, что таким образом введенные скалярные произведения удовлетворяют аксиомам 1—4.

Длина вектора арифметического пространства

$$|\mathbf{x}| = \sqrt{\sum_{i=1}^n |x_i|^2}.$$

В дальнейшем будем рассматривать конечномерное евклидово пространство E_n .

6. Ортогональные системы векторов. Два вектора $\mathbf{x}$ и $\mathbf{y}$ пространства E_n называются ортогональными, если их скалярное произведение равно нулю:

$$(\mathbf{x}, \mathbf{y}) = 0.$$

Система векторов $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ называется ортогональной, если любые два вектора системы ортогональны друг другу, т. е.

$$(\mathbf{x}_j, \mathbf{x}_k) = 0 \text{ при } j \neq k.$$

Заметим, что если вектор $\mathbf{x}_1$ ортогонален векторам $\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n$, то этот вектор ортогонален также любой линейной комбинации последних векторов, иными словами, вектор $\mathbf{x}_1$ ортогонален пространству, натянутому на векторы $\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n$. Ненулевые попарно ортогональные векторы $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ линейно независимы и, следовательно, в n -мерном пространстве E_n ортогональная система содержит не более n векторов.

Базис $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ пространства E_n называют ортогональным, если базисные векторы попарно ортогональны, т. е. $(\mathbf{e}_j, \mathbf{e}_k) = 0$ при $j \neq k$, $j, k = 1, 2, \dots, n$.

Если, кроме того, базисные векторы удовлетворяют условию $(\mathbf{e}_j, \mathbf{e}_j) = 1$, $j = 1, 2, \dots, n$, то ортогональный базис называют нормированным (ортонормированным). В этом случае имеем

$$(\mathbf{e}_j, \mathbf{e}_k) = \delta_{jk},$$

где δ_{jk} — символ Кронекера.

Например, в арифметическом пространстве ортонормированный базис образует система векторов

$$\mathbf{e}_1 = (1, 0, 0, \dots, 0),$$

$$\mathbf{e}_2 = (0, 1, 0, \dots, 0),$$

$$\mathbf{e}_n = (0, 0, 0, \dots, 1).$$

Ортогональный базис $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ всегда можно нормировать, разделив каждый из векторов $\mathbf{e}_j$ на его длину. Полученные новые векторы

$$\mathbf{e}_j = \frac{\mathbf{e}_j}{\sqrt{(\mathbf{e}_j, \mathbf{e}_j)}}$$

образуют ортонормированный базис.

Выразим координаты вектора $\mathbf{x}$ в ортонормированном базисе $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$. Если

$$\mathbf{x} = c_1 \mathbf{e}_1 + c_2 \mathbf{e}_2 + \dots + c_n \mathbf{e}_n, \quad (\text{II.5})$$

то, умножая скалярно равенство (II.5) справа на $\mathbf{e}_j$, получаем $c_j = (\mathbf{x}, \mathbf{e}_j)$, $j = 1, 2, \dots, n$, т. е. в ортонормированном базисе координата вектора равна скалярному произведению его на соответствующий базисный вектор. В этом случае считается, что координата вектора равна его проекции на соответствующий вектор базиса.

Умножив скалярно (II.5) на $\mathbf{x}$, получим

$$(\mathbf{x}, \mathbf{x}) = \left(\sum_{j=1}^n c_j \mathbf{e}_j, \sum_{k=1}^n c_k \mathbf{e}_k \right) = \sum_{j=1}^n \sum_{k=1}^n c_j c_k (\mathbf{e}_j, \mathbf{e}_k) = \sum_{j=1}^n c_j^2,$$

т. е. квадрат длины вектора равен сумме квадратов его проекций на базисные ортонормированные векторы.

7. Преобразование координат вектора при изменениях базиса. Пусть $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ и $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ — два базиса одного и того же линейного пространства E_n . Каждый вектор нового (второго) базиса $\mathbf{e}_j$, $j = 1, 2, \dots, n$, имеет в старом (первом) базисе $\mathbf{e}_i$ некоторые координаты $s_{1j}, s_{2j}, \dots, s_{nj}$, т. е.

$$\mathbf{e}_j = s_{1j} \mathbf{e}_1 + s_{2j} \mathbf{e}_2 + \dots + s_{nj} \mathbf{e}_n, \quad j = 1, 2, \dots, n. \quad (\text{II.6})$$

Неособенная матрица S с элементами s_{ij} , $j = 1, 2, \dots, n$, называется матрицей перехода от старого базиса к новому. Определитель $|S| \neq 0$, так как в противном случае векторы $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ были бы линейно зависимы. Установим связь между координатами одного и того же вектора в различных базисах. Пусть $\mathbf{x}$ — данный вектор. Обозначим через x_i координаты этого вектора в старом базисе и через ξ_j его координаты в новом базисе. Очевидно, что

$$\mathbf{x} = \sum_{i=1}^n x_i \mathbf{e}_i = \sum_{j=1}^n \xi_j \mathbf{e}_j.$$

Отсюда, подставляя во вторую сумму выражение (II.6), получаем

$$\mathbf{x} = \sum_{i=1}^n x_i \mathbf{e}_i = \sum_{j=1}^n \xi_j \sum_{i=1}^n s_{ij} \mathbf{e}_i = \sum_{i=1}^n \mathbf{e}_i \sum_{j=1}^n s_{ij} \xi_j.$$

Следовательно, в силу линейной независимости векторов $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ находим

$$x_i = \sum_{j=1}^n s_{ij} \xi_j, \quad i = 1, 2, \dots, n. \quad (\text{II.7})$$

⁵ При обозначении координат на первом месте указывается номер вектора старого базиса, а на втором — нового базисного вектора.

Формулы (II.13) дают полную систему решений уравнения (II.10). За фундаментальную систему решений можно принять

$$\mathbf{x}^{(1)} = \begin{bmatrix} \alpha_{11} \\ \vdots \\ \alpha_{r1} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{x}^{(2)} = \begin{bmatrix} \alpha_{12} \\ \vdots \\ \alpha_{r2} \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, \mathbf{x}^{(k)} = \begin{bmatrix} \alpha_{1k} \\ \vdots \\ \alpha_{rk} \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Эти решения могут быть найдены непосредственно из системы (II.12), если в ней последовательно полагать

$$\begin{array}{ccccccc} x_{r+1}=1, & x_{r+2}=x_{r+3}=\cdots=x_n=0, \\ x_{r+1}=0, & x_{r+2}=1, & x_{r+3}=\cdots=x_n=0, \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & x_{r+1}=\cdots=x_{n-1}=0, & x_n=1. \end{array}$$

9. Линейное преобразование переменных. Пусть $x_1, x_2, \dots, x_n$ — некоторая система переменных величин, а $y_1, y_2, \dots, y_n$ — другая система переменных величин, связанная с первой следующими соотношениями:

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n), \\ y_2 &= f_2(x_1, x_2, \dots, x_n), \\ &\vdots \\ y_n &= f_n(x_1, x_2, \dots, x_n), \end{aligned} \quad (\text{II.14})$$

где $f_1, f_2, \dots, f_n$ — заданные функции.

Переход от системы $x_1, x_2, \dots, x_n$ к системе $y_1, y_2, \dots, y_n$ называют преобразованием переменных. Преобразование (II.14) называется линейным, если новые переменные $y_1, y_2, \dots, y_n$ являются линейными однородными функциями старых переменных $x_1, x_2, \dots, x_n$, т. е.

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n, \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n, \\ &\vdots \\ y_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n, \end{aligned} \quad (\text{II.15})$$

где a_{ij} , $i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$, — постоянные.

Линейное преобразование (II.15) однозначно определяется своей квадратной матрицей коэффициентов (матрицей преобразования)

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}.$$

В то же время, имея квадратную матрицу A , можно построить линейное преобразование, для которого эта матрица служит матрицей коэффициентов. Таким образом, между линейными преобразованиями и квадратными матрицами существует взаимно-однозначное соответствие.

Системы переменных $x_1, x_2, \dots, x_n$ и $y_1, y_2, \dots, y_n$ можно рассматри-
вать как вектор-столбцы

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

одного и того же векторного пространства E_n . Поэтому преобразование (II.15) отображает пространство E_n само на себя или на свою правильную часть (если $|A| = 0$). Соотношение (II.15) эквивалентно одному матричному соотношению

$$\mathbf{y} = A\mathbf{x}. \quad (\text{II.16})$$

Таким образом, матрицу A можно рассматривать как линейный оператор (линейное преобразование). Линейный оператор обладает следующими свойствами: он аддитивен, т. е. для всех $x, y \in E_n$

$$A(\mathbf{x} + \mathbf{y}) = A\mathbf{x} + A\mathbf{y},$$

он однороден, т. е. для $x \in E_n$ и $\alpha \in K$

$$A(\alpha \mathbf{x}) = \alpha A\mathbf{x}.$$

Оператор A^* называется сопряженным с A , если для любых векторов $x, y \in E_n$ выполняется равенство

$$(A\mathbf{x}, \mathbf{y}) = (\mathbf{x}, A^*\mathbf{y}).$$

Оператор A называется самосопряженным, если он совпадает со своим сопряженным оператором A^* , т. е. $(Ax, y) = (x, Ay)$.

Введем следующие действия над линейными преобразованиями A и B : сложение $A + B$, умножение на скаляр αA , умножение BA с помощью соглашений

$$(A \div B) \mathbf{x} = A\mathbf{x} \div B\mathbf{x}, \quad (\text{II.17})$$

$$(\alpha A) \mathbf{x} = \alpha (A\mathbf{x}), \quad (\text{II.18})$$

$$(BA) \mathbf{x} = B(A\mathbf{x}), \quad (\text{II.19})$$

где x — вектор,

Если A и B — линейные преобразования, то результирующие преобразования $A + B$, αA , BA также линейные. Каждому действию (II.17) — (II.19) над линейными преобразованиями соответствует такое же действие над матрицами этих линейных преобразований.

Пусть линейное преобразование (II.15) переводит систему переменных $x_1, x_2, \dots, x_n$ в систему переменных $y_1, y_2, \dots, y_n$. Преобразование, переводящее систему переменных $y_1, \dots, y_n$ в систему переменных $x_1, x_2, \dots, x_n$, называется обратным по отношению к преобразованию (II.15).

Формулы обратного преобразования получают, разрешив систему (II.15) относительно переменных $x_1, x_2, \dots, x_n$. Линейное преобразование имеет однозначное обратное преобразование тогда и только тогда, когда матрица данного преобразования неособенная. Обратное преобразование линейного преобразования есть линейное, и его матрица является обратной относительно матрицы исходного преобразования.

Если преобразование A вырожденное ($|A| = 0$), то формула (II.16) преобразует пространство в подпространство меньшего числа измерений. Будем понимать под E тождественное преобразование, оставляющее вектор x неизменным.

10. Собственные векторы и собственные значения матриц. Пусть дана квадратная матрица A . Рассмотрим линейное преобразование (II.16). Ненулевой вектор называется собственным вектором данной матрицы или определяемого ею линейного преобразования, если в результате соответствующего линейного преобразования этот вектор переходит в вектор, отличающийся от исходного скалярным множителем. Иными словами, вектор $x \neq 0$ называется собственным вектором матрицы A , если эта матрица переводит вектор x в вектор λx , т. е.

$$Ax = \lambda x. \quad (\text{II.20})$$

Число λ в (II.20) называют собственным значением или характеристическим числом матрицы A , соответствующим данному собственному вектору x .

Уравнение (II.20) имеет ненулевое решение тогда и только тогда, когда определитель матрицы $A - \lambda E$ равен нулю:

$$|A - \lambda E| = 0. \quad (\text{II.21})$$

Этот определитель называют характеристическим определителем матрицы A , а уравнение (II.21) называют характеристическим уравнением матрицы A . В развернутом виде это характеристическое уравнение записывается следующим образом:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} - \lambda & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} - \lambda \end{vmatrix} = 0, \quad (\text{II.22})$$

или

$$(-\lambda)^n + \sigma_1(-\lambda)^{n-1} + \sigma_2(-\lambda)^{n-2} + \dots + \sigma_{n-1}(-\lambda) + \sigma_n = 0. \quad (\text{II.23})$$

Полином, стоящий в левой части уравнения (II.23), называется характеристическим полиномом матрицы A . Коэффициент σ_1 равен сумме диагональных элементов матрицы A . Коэффициент σ_2 есть сумма главных миноров второго порядка матрицы A . Вообще, коэффициент σ_k есть сумма всех главных миноров k -го порядка матрицы A . И, наконец, свободный член σ_n равен определителю матрицы A .

Уравнение (II.23) есть нелинейное алгебраическое уравнение n -й степени относительно λ и, следовательно, имеет, по крайней мере, один действительный или комплексный корень. Пусть $\lambda_1, \lambda_2, \dots, \lambda_m$,

$m \leq n$, — различные корни уравнения (II.23). Эти корни будут собственными значениями матрицы A , а совокупность всех собственных значений называют спектром матрицы A . При подстановке корня $\lambda = \lambda_j$ в уравнение (II.20) получают

$$(A - \lambda_j E)x = 0. \quad (\text{II.24})$$

Так как определитель системы (II.24) $|A - \lambda_j E| = 0$, то эта система заведомо имеет ненулевые решения, которые и являются собственными векторами матрицы A , соответствующими собственному значению λ_j . Если ранг матрицы $A - \lambda_j E$ равен r ($r < n$), то существуют $k = n - r$ линейно независимых собственных векторов

$$x_j^{(1)}, x_j^{(2)}, \dots, x_j^{(k)},$$

отвечающих корню λ_j .

Отметим, что число линейно независимых собственных векторов, отвечающих одному и тому же корню характеристического уравнения, не превышает кратности этого корня. Отсюда следует, что если корни характеристического уравнения (II.23) различны, то каждому собственному значению соответствует с точностью до коэффициента пропорциональности один и только один собственный вектор.

Собственные векторы матрицы, соответствующие попарно различным собственным значениям, линейно независимы.

Если все собственные значения матрицы A порядка n попарно различны, то отвечающие им n собственных векторов этой матрицы образуют базис соответствующего n -мерного пространства.

Если значения $\lambda_j, j = 1, 2, \dots, n$, являются собственными значениями матрицы A , то собственные значения матрицы $A - tE$ равны $\lambda_j - t$, а собственные векторы матриц A и $A - tE$ совпадают.

Пусть $\lambda_j, j = 1, 2, \dots, n$, — собственные значения матрицы A , тогда $\lambda_j^p, j = 1, 2, \dots, n$, являются собственными значениями матрицы A^p , причем собственные векторы матрицы A будут собственными векторами матрицы A^p .

Если матрица A невырожденная и имеет собственные значения $\lambda_j, j = 1, 2, \dots, n$, то собственные значения A^{-1} равны $\frac{1}{\lambda_j}, j = 1, 2, \dots, n$, а собственные векторы матриц A и A^{-1} одинаковы.

11. Подобные матрицы. Две матрицы, соответствующие одному и тому же линейному преобразованию в различных базисах, называют подобными. Пусть матрица A в некотором базисе реализует линейное преобразование

$$y = Ax. \quad (\text{II.25})$$

В новом базисе это же линейное преобразование будет описываться другой матрицей:

$$\eta = B\xi, \quad (\text{II.26})$$

где матрица B подобна A .

Обозначим через S матрицу перехода от новой системы координат к старой, т. е. пусть

$$x = S\xi, y = S\eta. \quad (\text{II.27})$$

Подставив (II.27) в (II.25), получим

$$S\eta = AS\xi.$$

Умножив слева последнее равенство на обратную матрицу S^{-1} , будем иметь

$$\eta = S^{-1}AS\xi. \quad (\text{II.28})$$

Сравнивая (II.28) и (II.26), получаем

$$B = S^{-1}AS. \quad (\text{II.29})$$

Преобразование (II.29) называется преобразованием подобия матрицы A с помощью матрицы S .

Таким образом, две матрицы подобны тогда и только тогда, когда одна получается из другой путем преобразования подобия с помощью некоторой неособенной матрицы.

Две подобные матрицы A и B имеют одинаковый характеристический полином и, следовательно, одинаковые собственные числа (включая их кратность), а собственные векторы матрицы A равны:

$$z_i = Sy_i,$$

где y_i — собственные векторы матрицы B .

Отметим, что всякую квадратную матрицу, собственные значения которой попарно различны, путем преобразования подобия можно привести к диагональному виду.

12. Билинейная и квадратичная формы матриц. Пусть A — фиксированная действительная квадратная матрица и x, y — произвольные векторы n -мерного действительного пространства. Составим скалярное произведение

$$(Ax, y) = \sum_{j=1}^n (Ax)_j y_j = \sum_{j=1}^n \sum_{k=1}^n a_{jk} x_k y_j. \quad (\text{II.30})$$

Выражение (II.30) называют билинейной формой матрицы A , которую можно переписать следующим образом:

$$(Ax, y) = \sum_{j=1}^n \sum_{k=1}^n a_{jk} y_j x_k = (A^T y, x) = (x, A^T y).$$

Таким образом, в билинейной форме (II.30) матрицу можно переставлять с первого места на второе, заменяя ее транспонированной.

Если матрица A симметричная ($A^T = A$), то

$$(Ax, y) = (x, Ay),$$

т. е. в скалярном произведении симметричную матрицу можно переставлять с первого места на второе.

Выражение

$$(Ax, x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$$

называют квадратичной формой матрицы.

13. Свойства симметричных матриц. Важным частным случаем матриц являются симметричные матрицы. Матрица самосопряженного оператора в любом ортонормированном базисе есть симметричная матрица. Для симметричных матриц справедливы ряд свойств, представляющих для нас определенный интерес: все собственные значения симметричной матрицы действительны; собственные векторы симметричной матрицы, соответствующие различным собственным значениям, ортогональны между собой; всякую симметричную матрицу с помощью преобразования подобия можно привести к диагональному виду; каждому собственному значению симметричной матрицы соответствует столько линейно независимых собственных векторов, какова кратность собственного значения; симметричная матрица является положительно определенной тогда и только тогда, когда все собственные значения ее положительны; в положительно полуопределенных матрицах все собственные значения неотрицательны; если A — симметричная матрица и

$$\delta = \min \{\lambda_1, \lambda_2, \dots, \lambda_n\}, \Delta = \max \{\lambda_1, \lambda_2, \dots, \lambda_n\},$$

где $\lambda_1, \dots, \lambda_n$ — собственные значения матрицы A , то для любого вектора x справедливы неравенства

$$\delta(x, x) \leq (Ax, x) \leq \Delta(x, x); \quad (\text{II.31})$$

спектральная норма симметричной матрицы равна максимальному из модулей собственных значений: если симметричные матрицы A и B перестановочны, т. е. $AB = BA$, то они имеют общую систему собственных векторов; если A и B — перестановочные симметричные матрицы ($AB = BA$), то матрица AB имеет ту же систему собственных векторов, что и BA , и ее собственные значения $\lambda_{AB}^{(k)}$ могут быть определены по формуле

$$\lambda_{AB}^{(k)} = \lambda_A^{(k)} \lambda_B^{(k)}, \quad k = 1, 2, \dots, n, \quad (\text{II.32})$$

где $\lambda_A^{(k)}$ и $\lambda_B^{(k)}$ — собственные значения соответственно матриц A и B , отвечающие одному собственному вектору.

Для суммы этих матриц справедливо равенство

$$\lambda_{A+B}^{(k)} = \lambda_A^{(k)} + \lambda_B^{(k)}, \quad k = 1, 2, \dots, n. \quad (\text{II.33})$$

II.2. ОДНОШАГОВЫЕ ИТЕРАЦИОННЫЕ ПРОЦЕССЫ

1. Предварительные замечания. При решении систем линейных алгебраических уравнений с матрицами большого порядка, обладающими определенной спецификой (ленточного вида, легко программно порождающимися), реализация прямых методов, даже учитывающих разреженность матрицы, на ЭВМ становится затруднительной. Трудности машинной реализации прямых методов объясняются ограничением объема оперативного запоминающего устройства и снижением быстродействия при использовании памяти второй ступени. Запись систем с матрицами, обладающими определенной спецификой, в виде процедуры вычисления вектора Ax или сохранение программы порождения

элементов матрицы требует в ряде случаев меньшей памяти, чем хранение разреженной матрицы и вектора правых частей, необходимых при реализации прямых методов. В этих случаях для решения систем линейных алгебраических уравнений на ЭВМ целесообразно использовать быстроходящиеся итерационные методы.

Итерационные методы являются приближенными. Они дают решение системы как предел последовательных приближений, вычисляемых некоторым единообразным процессом. Вычислительные схемы итерационных методов просты и удобны в реализации на ЭВМ, экономят память машины и позволяют исключать действия с нулевыми элементами матриц. Быстроходящиеся итерационные процессы за счет некоторого снижения точности позволяют получить машинное решение за меньшее по сравнению с прямыми методами количество арифметических действий.

Однако каждый итерационный процесс характеризуется своими условиями сходимости, практическая проверка которых должна предшествовать применению используемого метода. Построение быстроходящихся итерационных процессов требует дополнительной информации о матрице, которую в случае матриц плотного заполнения получить затруднительно.

В общем случае сопоставление объема требуемой памяти ЭВМ, числа арифметических операций, длины машинного слова, времени решения на конкретной ЭВМ позволяет сделать вывод о целесообразности применения для конкретной системы прямого или итерационного метода решения.

2. Итерационный метод Якоби (простая итерация). Построение и применение итерационных методов рассмотрим на примере метода Якоби. При описании некоторых прикладных задач встречаются системы

$$Ax = f, \det A \neq 0, \quad (\text{II.34})$$

с квадратными матрицами, диагональные элементы которых преобладают над остальными.

В системе (II.34)

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}.$$

Для применения итерационного метода систему (II.34) перепишем следующим образом:

$$x = Mx + \varphi, \quad (\text{II.35})$$

где

$$M = D^{-1}(D - A), \quad \varphi = D^{-1}f, \\ D = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix},$$

$$M = \begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \dots & -\frac{a_{2n}}{a_{22}} \\ \dots & \dots & \dots & \dots & \dots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \dots & 0 \end{bmatrix}, \quad \varphi = \begin{bmatrix} \frac{f_1}{a_{11}} \\ \frac{f_2}{a_{22}} \\ \vdots \\ \frac{f_n}{a_{nn}} \end{bmatrix}.$$

Выберем произвольное начальное приближение $x^0 = x_0$ и построим векторы

$$\begin{aligned} x^{(1)} &= Mx^{(0)} + \varphi, \\ x^{(2)} &= Mx^{(1)} + \varphi, \\ &\dots \dots \dots \\ x^{(k+1)} &= Mx^{(k)} + \varphi. \end{aligned} \quad (\text{II.36})$$

Если последовательность $x^{(0)}, x^{(1)}, \dots, x^{(k+1)}, \dots$ имеет предел x , то он является решением системы (II.35). Действительно, переходя в равенстве (II.36) к пределу при $k \rightarrow \infty$, получаем (II.35), что и доказывает наше утверждение.

Итерационные процессы, в которых $(k+1)$ -е приближение зависит только от k -го приближения, называются одношаговыми.

Для сходимости процесса итераций при любом начальном векторе $x^{(0)}$ и при любом значении φ необходимо и достаточно, чтобы все собственные числа матрицы M были по модулю меньше единицы.

Легко убедиться в том, что

$$x^{(k+1)} = M^{k+1}x^{(0)} + (E + M + \dots + M^k)\varphi. \quad (\text{II.37})$$

Действительно, при $k=1$ это не вызывает сомнения, а для остальных k проверяется методом полной индукции. Отсюда следует, что для сходимости итерационного процесса для любого φ необходимо, чтобы сошелся ряд из матриц

$$E + M + \dots + M^k + \dots = \sum_{k=0}^{\infty} M^k. \quad (\text{II.38})$$

В [103] показано, что ряд (II.38) сходится, если все собственные значения $\lambda_j, j=1, 2, \dots, n$, матрицы M удовлетворяют неравенствам

$$|\lambda_j| < 1, \quad j=1, 2, \dots, n. \quad (\text{II.39})$$

Так как при этом $M^k \rightarrow 0$ при $k \rightarrow \infty$, то из формулы (II.37) вытекает, что процесс итераций сходится при любых φ и $x^{(0)}$, т. е. существует

$$\lim_{k \rightarrow \infty} x^{(k+1)} = x,$$

где x — решение системы (II.35). Если неравенства (II.39) не выполняются, то ряд (II.38), следовательно, и процесс итераций расходятся.

Таким образом, для сходимости процесса итераций (II.36) необходимо и достаточно, чтобы все собственные значения по модулю были

меньше единицы. Для проверки необходимых и достаточных условий можно использовать степенной метод нахождения максимального собственного значения матрицы (см. п. 2 параграфа III.4). Однако это требует дополнительных затрат машинного времени, поэтому иногда можно использовать достаточные признаки сходимости, непосредственно связанные с элементами матрицы.

Для сходимости процесса итераций достаточно, чтобы какая-либо норма матрицы M была меньше единицы. Действительно, если $\|M\| < 1$, то все собственные числа матрицы меньше единицы, и поэтому на основании предыдущих рассуждений итерационный процесс сходится.

Однако сходимость итерационного процесса не гарантирует близости итерационного решения $x^{(k)}$ к математическому решению x^* системы (II.34) при конечных k . Поэтому возникает необходимость сформулировать такие условия окончания итерационного процесса, которые обеспечивают достаточную близость итерационного и математического решений задачи.

Выполнение условия

$$\frac{\|r^{(k)}\|}{\|f\|} \leq \frac{\varepsilon}{\text{cond } A} \quad (\text{II.40})$$

(где $r^{(k)} = Ax^{(k)} - f$; $\text{cond } A = \|A\| \|A^{-1}\|$; ε — как угодно малое наперед заданное число) обеспечивает выполнение неравенства

$$\frac{\|x^{(k)} - x^*\|}{\|x^*\|} \leq \varepsilon, \quad \|x^*\| \neq 0, \quad (\text{II.41})$$

Действительно, из неравенств

$$\|f\| \leq \|A\| \|x^*\|, \quad \|x^{(k)} - x^*\| \leq \|A^{-1}\| \|r^{(k)}\|$$

получаем

$$\frac{\varepsilon}{\text{cond } A} \geq \frac{\|r^{(k)}\|}{\|f\|} = \frac{\|r^{(k)}\|}{\|Ax^*\|} \geq \frac{\|x^{(k)} - x^*\|}{\|x^*\|},$$

откуда и следует неравенство (II.41) [67].

Вычислительная схема метода может быть реализована следующим образом. Выбирают в качестве начального приближения к искомому решению x^* произвольный вектор $x^{(0)} = x_0$ и задают ε . Затем для каждого $i = 1, 2, \dots, n$ вычисляют последовательные приближения по формулам

$$x_i^{(k+1)} = \frac{f_i}{a_{ii}} - \sum_{j=1, j \neq i}^n \frac{a_{ij}}{a_{ii}} x_j^{(k)} \quad (\text{II.42})$$

и проверяют на каждой итерации условие окончания итераций (II.40). Итерационный процесс (II.42) можно также оканчивать при выполнении неравенства

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \leq \frac{\varepsilon}{1 + \varepsilon} \frac{1}{\|A^{-1}\| \|D\|}, \quad \|x^{(k)}\| \neq 0. \quad (\text{II.43})$$

Это условие гарантирует выполнение неравенства (II.41). Итерации по формулам (II.42) повторяют до тех пор, пока не будет выполнено условие их окончания.

В каждом итерационном процессе целесообразно оценивать объем вычислительной работы для получения решения системы с заданной точностью. Рассмотрим один из возможных способов оценки числа итераций.

Если $\|M\| = \mu < 1$, то справедлива оценка

$$\|x^{(k)} - x^*\| \leq \frac{\mu^k}{1 - \mu} \|x^{(0)} - Mx^{(0)} - \varphi\|, \quad (\text{II.44})$$

где x^* — точное решение системы (II.34). Действительно,

$$\|x^{(k)} - x^*\| = \|Mx^{(k-1)} + \varphi - Mx^* - \varphi\| < \mu \|x^{(k-1)} - x^*\|,$$

т. е.

$$\|x^{(k)} - x^*\| \leq \mu^k \|x^{(0)} - x^*\|.$$

Далее,

$$\begin{aligned} \|x^{(0)} - x^*\| &= \|x^{(0)} - Mx^{(0)} + Mx^{(0)} - Mx^* - \varphi\| \leq \|x^{(0)} - Mx^{(0)} - \varphi\| + \\ &+ \|Mx^{(0)} - Mx^*\| \leq \|x^{(0)} - Mx^{(0)} - \varphi\| + \mu \|x^{(0)} - x^*\|, \end{aligned}$$

т. е.

$$\|x^{(0)} - x^*\| < \frac{1}{1 - \mu} \|x^{(0)} - Mx^{(0)} - \varphi\|.$$

Используя последнее неравенство для оценки $\|x^{(k)} - x^*\|$, получаем (II.44).

В то же время справедлива оценка

$$\|r^{(k)}\| = \|f - Ax^{(k)}\| = \|Ax^* - Ax^{(k)}\| = \|A(x^* - x^{(k)})\| \leq \|A\| \|x^{(k)} - x^*\|,$$

т. е.

$$\|r^{(k)}\| \leq \|A\| \|x^{(k)} - x^*\|. \quad (\text{II.45})$$

Из (II.45) и (II.44) следует

$$\frac{\|r^{(k)}\|}{\|f\|} \leq \frac{\mu^k}{1 - \mu} \frac{\|A\|}{\|f\|} \|x^{(0)} - Mx^{(0)} - \varphi\|.$$

Определим теперь число итераций k из условия

$$\frac{\mu^k}{1 - \mu} \frac{\|A\|}{\|f\|} \|x^{(0)} - Mx^{(0)} - \varphi\| \leq \frac{\varepsilon}{\text{cond } A}.$$

Из этого соотношения имеем

$$\mu^k \leq \frac{\varepsilon (1 - \mu) \|f\|}{\text{cond } A \|A\| \|x^{(0)} - Mx^{(0)} - \varphi\|},$$

следовательно,

$$k \geq \frac{1}{\ln \mu} \ln \frac{\varepsilon (1 - \mu) \|f\|}{\text{cond } A \|A\| \|x^{(0)} - Mx^{(0)} - \varphi\|}.$$

На одну итерацию в этом методе требуется n операций деления, $n^2 - n$ операций умножения, n^2 операций сложения.

Пример 1. Итерационным методом Якоби решить систему линейных алгебраических уравнений

$$\begin{aligned} 12x_1 - 3x_2 + x_3 &= 9, \\ x_1 + 5x_2 - x_3 &= 8, \\ x_1 - x_2 + 3x_3 &= 8 \end{aligned} \quad (\text{II.46})$$

с точностью $\epsilon = 10^{-2}$.

Решение. Запишем систему (II.46) в виде (II.35)

$$\begin{aligned} x_1 &= \frac{3}{4} + \frac{1}{4}x_2 - \frac{1}{12}x_3, \\ x_2 &= \frac{8}{5} - \frac{1}{5}x_1 + \frac{1}{5}x_3, \\ x_3 &= \frac{8}{3} - \frac{1}{3}x_1 + \frac{1}{3}x_2 \end{aligned}$$

и проверим достаточные условия применимости итерационного процесса. Нетрудно убедиться, что норма матрицы ($\|M\|_1$ или $\|M\|_\infty$)

$$M = \begin{bmatrix} 0 & \frac{1}{4} & -\frac{1}{12} \\ -\frac{1}{5} & 0 & \frac{1}{5} \\ -\frac{1}{3} & \frac{1}{3} & 0 \end{bmatrix}$$

меньше единицы. Таким образом, для системы (II.46) выполнены достаточные условия сходимости итерационного процесса (II.36).

Задавая нулевым начальным приближением, вычисляем $x^{(k+1)}$ по формулам (II.42), проверяя условие окончания итераций (II.40). В результате получаем приближение к решению системы (II.46), которое приведено в табл. 2. Использование вместо (II.40) условия (II.43) при $\epsilon = 10^{-2}$ приводит к останову процесса после пятой итерации и к следующим результатам: $x_1^{(5)} = 1,00080$; $x_2^{(5)} = 2,00040$; $x_3^{(5)} = 3,00140$.

Т а б л и ц а 2

$x^{(k)}$	k				
	0	1	2	3	4
$x_1^{(k)}$	0	0,750 000	0,927 783	0,999 991	0,999 566
$x_2^{(k)}$	0	1,600 00	1,983 32	2,004 46	2,003 72
$x_3^{(k)}$	0	1,983 32	2,950 00	3,018 51	3,001 50

Метод итераций Якоби находит применение при решении систем линейных алгебраических уравнений с несимметричными матрицами.

3. Принцип построения итерационных процессов [77]. Подставляя значение M в (II.36), получаем итерационную формулу в следующем виде:

$$x^{(k+1)} = D^{-1}(D - A)x^{(k)} + D^{-1}f, \quad x^{(0)} = x_0.$$

Нетрудно убедиться, что эту формулу можно переписать так:

$$Dx^{(k+1)} = (D - A)x^{(k)} + f,$$

или

$$D(x^{(k+1)} - x^{(k)}) + Ax^{(k)} = f, \quad x^{(0)} = x_0. \quad (\text{II.47})$$

Запись итерационной схемы (II.36) в виде (II.47) является частным случаем записи итерационных процессов в канонической форме

$$B_k(x^{(k+1)} - x^{(k)}) + Ax^{(k)} = f, \quad x^{(0)} = x_0, \quad (\text{II.48})$$

где B_k ($k = 0, 1, 2, \dots$) — некоторая последовательность невырожденных матриц, k — номер итерации, x_0 — произвольное начальное приближение. Различный выбор последовательностей матриц B_k приводит к различным итерационным процессам.

Итерационные процессы, записанные в канонической форме (II.48), обладают тем свойством, что для каждого из них точное решение x^* является неподвижной точкой. Это значит, что если за начальное приближение $x^{(0)}$ взять x^* , то все последующие приближения будут также равны x^* .

Всякий итерационный процесс, для которого x^* является неподвижной точкой, реализуемый по формуле

$$x^{(k+1)} = S^{(k)}x^{(k)} + q^{(k)}, \quad x^{(0)} = x_0, \quad (\text{II.49})$$

где $S^{(k)}$ — последовательность матриц, $q^{(k)}$ — последовательность векторов, может быть представлен в виде (II.48). Действительно, если x^* есть точное решение системы (II.34), то

$$x^* = S^{(k)}x^* + q^{(k)}.$$

Подставляя полученное значение $q^{(k)}$ в (II.49), имеем

$$x^{(k+1)} = S^{(k)}x^{(k)} + (E - S^{(k)})x^*.$$

Это равенство можно переписать следующим образом:

$$x^{(k+1)} = x^{(k)} + (E - S^{(k)})x^* - x^{(k)} + S^{(k)}x^{(k)},$$

или

$$x^{(k+1)} = x^{(k)} + (E - S^{(k)})x^* - (E - S^{(k)})x^{(k)}.$$

Отсюда следует, что

$$x^{(k+1)} = x^{(k)} + (E - S^{(k)})x^* - (E - S^{(k)})A^{-1}Ax^{(k)},$$

или

$$x^{(k+1)} = x^{(k)} + (E - S^{(k)})A^{-1}f - (E - S^{(k)})A^{-1}Ax^{(k)},$$

и, наконец,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + (E - S^{(k)}) A^{-1} (\mathbf{f} - A\mathbf{x}^{(k)}).$$

Из последнего равенства легко получается каноническая форма (II.48)

$$B_k (\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) + A\mathbf{x}^{(k)} = \mathbf{f}, \quad \mathbf{x}^{(0)} = \mathbf{x}_0,$$

где

$$B_k = A(E - S^{(k)})^{-1}.$$

Сформулируем необходимые и достаточные условия сходимости итерационных процессов, записанных в канонической форме (II.48). Введем в рассмотрение погрешность k -го приближения к решению $\mathbf{z}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}^*$. Нетрудно убедиться, что формулу (II.48) относительно погрешности $\mathbf{z}^{(k)}$ можно записать следующим образом:

$$B_k (\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}) + A\mathbf{z}^{(k)} = 0. \quad (\text{II.50})$$

Действительно, вычитая из (II.48) всегда выполняемое равенство

$$B_k (\mathbf{x}^* - \mathbf{x}^*) + A\mathbf{x}^* = \mathbf{f},$$

получаем (II.50). Уравнение (II.50) разрешим относительно $\mathbf{z}^{k+1}$:

$$\mathbf{z}^{(k+1)} = (E - B_k^{-1}A) \mathbf{z}^{(k)}.$$

Это равенство можно переписать так:

$$\begin{aligned} \mathbf{z}^{(k+1)} &= (E - B_k^{-1}A) \mathbf{z}^{(k)} = (E - B_k^{-1}A)(E - B_{k-1}^{-1}A) \mathbf{z}^{(k-1)} = \\ &= \prod_{i=0}^k (E - B_i^{-1}A) \mathbf{z}^{(0)} = T_k \mathbf{z}^{(0)}. \end{aligned} \quad (\text{II.51})$$

Для того чтобы погрешность $\mathbf{z}^{(k)}$ стремилась к нулю при любом начальном приближении $\mathbf{x}^{(0)} = \mathbf{z}^{(0)} + \mathbf{x}^*$, необходимо и достаточно, чтобы матрица T_k стремилась к нулевой [101], для чего, в свою очередь, достаточно, чтобы любая норма $\|T_k\|$ стремилась к нулю. Выведенное условие дает лишь общую точку зрения для построения условий сходимости конкретных итерационных процессов. Из рассмотрения формулы (II.51) ясно, что ускорить сходимость итераций можно лишь с помощью соответствующего выбора B_k , так как

$$T_k = \prod_{i=0}^k (E - B_i^{-1}A).$$

Для конкретного вида A строить B_k нужно таким образом, чтобы сходимость $\|T_k\|$ к нулю была наиболее быстрой.

Простейшими среди итерационных процессов являются стационарные итерационные процессы, в которых матрица B_k не зависит от номера шага k . В частности, при $B_k = D$ получается итерационный метод Якоби (см. п. 2 настоящего параграфа). Близкими к стационарным итерационным процессам являются циклические, в которых матрица B_k периодически повторяется через некоторое число p шагов. В нестационарных процессах изменение B_k осу-

ществляется на каждом шаге итераций. Выбор матрицы B для стационарного итерационного процесса и матрицы B_k для нестационарного процесса может осуществляться различными способами и на основании разных принципов.

Одним из принципов построения итерационных процессов является принцип релаксации. Под этим понимается принцип выбора матрицы B_k из некоторого класса матриц так, чтобы на каждом шаге процесса уменьшалась какая-либо величина, характеризующая точность приближения. Среди релаксационных методов различают координатные, в которых матрица B_k подбирается так, что на каждом шаге меняется одна или несколько компонент приближения, и градиентные, в которых матрицы B_k являются скалярными.

В качестве величины, характеризующей точность приближения, может выступать вектор ошибки $\mathbf{z} = \mathbf{x} - \mathbf{x}^*$ или вектор невязки $\mathbf{r} = \mathbf{f} - A\mathbf{x}$. Для положительно определенных матриц A удобной мерой точности является так называемая функция ошибки

$$F(\mathbf{x}) = (A\mathbf{z}, \mathbf{z}) = (\mathbf{z}, \mathbf{r}) = (A^{-1}\mathbf{r}, \mathbf{r}).$$

В силу положительной определенности A всегда $F(\mathbf{x}) \geq 0$, причем $F(\mathbf{x}) = 0$ только при $\mathbf{x} = \mathbf{x}^*$. Ясно, что

$$\begin{aligned} F(\mathbf{x}) &= (\mathbf{x}^* - \mathbf{x}, \mathbf{f} - A\mathbf{x}) = (\mathbf{x}^*, \mathbf{f}) - (\mathbf{x}, \mathbf{f}) - (\mathbf{x}^*, A\mathbf{x}) + (\mathbf{x}, A\mathbf{x}) = \\ &= (A\mathbf{x}, \mathbf{x}) - 2(\mathbf{x}, \mathbf{f}) + (\mathbf{x}^*, \mathbf{f}). \end{aligned}$$

Функция ошибки не может быть вычислена, если не известно точное решение. Однако ее значение лишь постоянным слагаемым отличается от значения функционала

$$F_0(\mathbf{x}) = (A\mathbf{x}, \mathbf{x}) - 2(\mathbf{x}, \mathbf{f}),$$

поэтому можно оценивать убывание функции ошибки, сравнивая соответствующие значения функционала $F_0(\mathbf{x})$.

Важным принципом построения итерационных процессов является принцип последовательного «подавления компонент» вектора ошибки в разложении его по собственным векторам матрицы коэффициентов системы. Рассмотрим, как решаются вопросы сходимости для систем линейных алгебраических уравнений с симметричными матрицами.

4. Некоторые итерационные методы решения систем линейных алгебраических уравнений с симметричными и положительно определенными матрицами. Рассмотрим систему линейных алгебраических уравнений

$$A\mathbf{y} = \mathbf{f} \quad (\text{II.52})$$

с симметричной и положительно определенной матрицей $A = A^T$. Для решения системы (II.52) можно использовать итерационный процесс, который является частным случаем процесса (II.48) и записывается в виде

$$B \frac{\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}}{\tau} + A\mathbf{y}^{(k)} = \mathbf{f}, \quad \mathbf{y}^{(0)} = \mathbf{y}_0, \quad (\text{II.53})$$

где B — симметричная и положительно определенная матрица, τ — итерационный параметр.

Итерационный процесс, который может быть записан в канонической форме (II.53) при

$$\tau = \frac{2}{\gamma_1 + \gamma_2}, \quad (\text{II.54})$$

где γ_1 и γ_2 — константы эквивалентности в неравенствах ⁶

$$\gamma_1 (By, y) \leq (Ay, y) \leq \gamma_2 (By, y), \quad (\text{II.55})$$

сходится к единственному решению системы (II.52) со скоростью геометрической прогрессии [85, 87], знаменатель которой

$$\rho = \frac{\gamma_2 - \gamma_1}{\gamma_2 + \gamma_1}. \quad (\text{II.56})$$

Итерационную схему (II.53) можно представить в виде

$$\frac{x^{(k+1)} - x^{(k)}}{\tau} + Cx^{(k)} = g, \quad x^{(0)} = B^{1/2}y_0, \quad (\text{II.57})$$

где $x^{(k)} = B^{1/2}y^{(k)}$, $g = B^{-1/2}f$, $C = B^{-1/2}AB^{-1/2}$, $B^{1/2}$ — квадратный корень из матрицы ⁷ B . Такое представление возможно, так как B положительно определенная матрица, и поэтому теоретически существуют положительно определенные матрицы $B^{1/2}$ и $B^{-1/2}$.

Введем в рассмотрение погрешность $z^{(k)} = x^{(k)} - x^*$, где x^* — точное решение системы

$$Cx = g, \quad (\text{II.58})$$

а $x^{(k)}$ — приближение к решению системы уравнений (II.58) на k -й итерации, $k = 0, 1, 2, \dots$. Тогда каноническую форму (II.57) относительно погрешности $z^{(k)}$ можно записать следующим образом:

$$\frac{z^{(k+1)} - z^{(k)}}{\tau} + Cz^{(k)} = 0, \quad z^{(0)} = x^{(0)} - x^*,$$

или

$$z^{(k+1)} = (E - \tau C) z^{(k)}, \quad z^{(0)} = x^{(0)} - x^*, \quad (\text{II.59})$$

где E — единичная матрица.

Так как C симметричная и положительно определенная матрица, то она обладает полной ортогональной системой собственных векторов $\{\psi_j\}$, $j = 1, 2, \dots, n$, соответствующих собственным значениям

$$0 < \gamma_1 \leq \mu_1 \leq \mu_2 \leq \dots \leq \mu_n \leq \gamma_2.$$

Разложим z^{k+1} по собственным векторам $\{\psi_j\}$, т. е. представим z^{k+1} суммой вида

$$z^{(k+1)} = \sum_{j=1}^n c_j^{(k+1)} \psi_j, \quad k = 0, 1, \dots \quad (\text{II.60})$$

⁶ Заметим, что матрицы A и B , удовлетворяющие неравенствам (II.55), называются энергетически эквивалентными, а сами константы γ_1 и γ_2 — константами энергетической эквивалентности.

⁷ Квадратный корень из положительно определенной матрицы B — это такая матрица S , что $S^2 = B$.

Подставим это представление в (II.59), получим

$$\sum_{j=1}^n c_j^{(k+1)} \psi_j = \sum_{j=1}^n (1 - \tau \mu_j) c_j^{(k)} \psi_j. \quad (\text{II.61})$$

Учитывая ортогональность системы векторов $\{\psi_j\}$, вместо (II.61) можно записать

$$c_j^{(k+1)} = (1 - \tau \mu_j) c_j^{(k)} = (1 - \tau \mu_j)^k c_j^{(0)}, \quad j = 1, 2, \dots, n. \quad (\text{II.62})$$

Поскольку все $\mu_j > 0$, $j = 1, 2, \dots, n$, то, как нетрудно убедиться, для сходимости итерационного процесса, т. е. для выполнения условия

$$|1 - \tau \mu_j| < 1, \quad j = 1, 2, \dots, n,$$

достаточно, чтобы

$$0 < \tau < \frac{2}{\mu_n}.$$

Рис. 4.

Для построения оптимального в некотором смысле итерационного процесса необходимо выбрать такое значение τ , которое минимизирует

$$\max_{1 \leq j \leq n} |1 - \tau \mu_j|, \quad 0 < \gamma_1 \leq \mu_j \leq \gamma_2. \quad (\text{II.63})$$

Для определения значения τ , доставляющего минимум (II.63), рассмотрим линейную функцию $|1 - \tau \mu|$, заданную на отрезке $[\gamma_1, \gamma_2]$. Известно, что линейная функция принимает свои экстремальные значения на концах отрезка. Значение τ , для которого величина $\max\{|1 - \tau \gamma_1|, |1 - \tau \gamma_2|\}$ наименьшая, получаем из соотношения

$$|1 - \tau \gamma_1| = |1 - \tau \gamma_2|. \quad (\text{II.64})$$

Действительно, нетрудно видеть (см. рис. 4), что изменение τ приводит к изменениям значения $|1 - \tau \mu|$ на концах отрезка $[\gamma_1, \gamma_2]$. Приравняв значения функции на концах отрезка, получаем искомое значение τ .

Таким образом, из (II.64) следует

$$1 - \tau \gamma_1 = 1 - \tau \gamma_2, \quad \text{или} \quad 1 - \tau \gamma_1 = -(1 - \tau \gamma_2).$$

Первое равенство справедливо при $\tau = 0$ или $\gamma_1 = \gamma_2$, что противоречит условию. Из второго равенства находим

$$\tau = \frac{2}{\gamma_1 + \gamma_2}.$$

При таком выборе τ для всех $j = 1, 2, \dots, n$

$$\min_{\tau} \max_{\gamma_1 \leq \mu_j \leq \gamma_2} |1 - \tau \mu_j| = \frac{\gamma_2 - \gamma_1}{\gamma_2 + \gamma_1} = \rho < 1. \quad (\text{II.65})$$

На основании разложения z по собственным векторам и формулы (II.62) можно записать

$$\|z^{(k)}\| = \left\| \sum_{i=1}^n c_i^{(k)} \psi_i \right\| \leq \rho \|z^{(k-1)}\| \leq \rho^k \|z^{(0)}\|. \quad (\text{II.66})$$

Следовательно, процесс уменьшения погрешности происходит со скоростью геометрической прогрессии, знаменатель которой равен ρ .

Таким образом, рассматриваемый нами итерационный процесс, записанный в канонической форме (II.53), при $\tau = \frac{2}{\gamma_1 + \gamma_2}$ теоретически сходится к решению задачи (II.52).

Для получения решения системы уравнений (II.52) с заданной «относительной» погрешностью ε , т. е. для получения неравенства (II.41), итерационный процесс можно закончить при выполнении условия (II.40) или условия

$$\frac{\|y^{(k+1)} - y^{(k)}\|}{\|y^{(k)}\|} \leq \frac{\varepsilon}{1 + \varepsilon} \frac{\tau}{\|A^{-1}\| \|B\|}, \quad \|y^{(k)}\| \neq 0, \quad (\text{II.67})$$

которое также обеспечивает выполнение условия (II.41).

Оценим количество итераций, которое необходимо для выполнения неравенства (II.40).

С учетом замены $y = B^{-1/2} x$ оценим норму невязки $r^{(k)}$:

$$\begin{aligned} \|r^{(k)}\| &= \|f - Ay^{(k)}\| = \|Ay - Ay^{(k)}\| \leq \|A\| \|y - y^{(k)}\| = \\ &= \|A\| \|B^{-1/2} (x - x^{(k)})\| \leq \|A\| \|B^{-1/2}\| \|z^{(k)}\|. \end{aligned} \quad (\text{II.68})$$

Из (II.66) и (II.68) следует

$$\|r^{(k)}\| \leq \rho^k \|A\| \|B^{-1/2}\| \|z^{(0)}\|. \quad (\text{II.69})$$

Учитывая, что $x^{(k+1)} = Sx^{(k)} + g$, где $S = E - \tau C$, и что $\|S\| = \rho$, имеем оценку

$$\begin{aligned} \|z^{(0)}\| &= \|x^{(0)} - x\| = \|x^{(0)} - Sx^{(0)} + Sx^{(0)} - Sx - g\| \leq \\ &\leq (1 - \rho) \|x^{(0)}\| + \|g\| + \rho \|z^{(0)}\|, \end{aligned}$$

откуда

$$\|z^{(0)}\| \leq \|x^{(0)}\| + \frac{1}{1 - \rho} \|g\|, \quad (\text{II.70})$$

но так как

$$\begin{aligned} \|x^{(0)}\| &= \|B^{1/2} y^{(0)}\| \leq \|B^{1/2}\| \|y^{(0)}\|, \\ \|g\| &= \|B^{-1/2} f\| \leq \|B^{-1/2}\| \|f\|, \end{aligned} \quad (\text{II.71})$$

то из (II.69), (II.70), (II.71) следует оценка

$$\|r^{(k)}\| \leq \rho^k \|A\| \|B^{-1/2}\| \left(\|B^{1/2}\| \|y^{(0)}\| + \frac{\|B^{-1/2}\|}{1 - \rho} \|f\| \right). \quad (\text{II.72})$$

Так как по предположению B — симметричная положительно определенная матрица, для нее верна оценка

$$\delta \|y\|^2 \leq (By, y) \leq \Lambda \|y\|^2, \quad \delta > 0,$$

и поэтому можно записать

$$\|A\| \leq \gamma_2 \Lambda, \quad \|A^{-1}\| \leq \frac{1}{\gamma_1 \delta}, \quad (\text{II.73})$$

$$\|B^{1/2}\| \leq \sqrt{\Lambda}, \quad \|B^{-1/2}\| \leq \sqrt{\frac{1}{\delta}}. \quad (\text{II.74})$$

Следовательно, учитывая изложенное выше, можно получить вместо (II.72) оценку

$$\|r^{(k)}\| \leq \rho^k \gamma_2 \Lambda \left(\sqrt{\frac{\Lambda}{\delta}} \|y^{(0)}\| + \frac{\|f\|}{\delta(1 - \rho)} \right). \quad (\text{II.75})$$

При выполнении (II.73) справедлива оценка

$$\text{cond } A \leq \frac{\gamma_2}{\gamma_1} \frac{\Lambda}{\delta},$$

поэтому из (II.74), (II.75) следует, что условие (II.41) будет выполнено, если

$$\rho^k \leq \bar{\varepsilon}, \quad \bar{\varepsilon} = \frac{\varepsilon \|f\| \gamma_1}{\gamma_2 \Lambda^2 \left[\sqrt{\frac{\Lambda}{\delta}} \|y^{(0)}\| + \frac{\|f\|}{\delta(1 - \rho)} \right]}. \quad (\text{II.76})$$

Из (II.76) получим оценку числа итераций

$$k \geq \frac{\ln \varepsilon}{\ln \rho}, \quad (\text{II.77})$$

где $\bar{\varepsilon}$ определено в (II.76).

Иногда трудно получить константу эквивалентности γ_1 в неравенствах (II.55). Тогда вместо оценки снизу γ_1 может быть взято любое число

$$\bar{\gamma}_1 = \gamma_1 + \alpha < \gamma_2, \quad \text{где } 0 \leq \alpha \leq \mu_n \leq \gamma_2.$$

В этом случае вместо параметра τ в (II.54) можно использовать параметр

$$\bar{\tau} = \frac{2}{\bar{\gamma}_1 + \gamma_2} = \frac{1}{\gamma_1 + \alpha + \gamma_2} = \frac{2}{\gamma_1 + \bar{\gamma}_2}, \quad \bar{\gamma}_2 = \gamma_2 + \alpha.$$

Повторяя предыдущие рассуждения, можно показать, что если матрицы A и B симметричны, положительно определены и вместо итерационного параметра τ используется параметр $\bar{\tau}$, то итерационные процессы, записываемые в канонической форме (II.53), сходятся к единственному решению системы (II.52) со скоростью геометрической прогрессии, знаменатель которой

$$\bar{\rho} = \frac{\bar{\gamma}_2 - \gamma_1}{\gamma_2 + \gamma_1}.$$

Естественно, что в этом случае $\bar{\rho} > \rho$ и скорость сходимости итераций при этом замедляется [121].

Если в канонической форме (II.53) $B = E$, то получаем явный одношаговый итерационный процесс

$$\frac{y^{(k+1)} - y^{(k)}}{\tau} + Ay^{(k)} = f, \quad y^{(0)} = y_0, \quad (\text{II.78})$$

в котором τ вычисляют по формуле

$$\tau = \frac{2}{\delta + \Delta}, \quad (\text{II.79})$$

где $\delta(y, y) \leq (Ay, y) \leq \Delta(y, y)$. Этот исследованный процесс (см., например, [29, 60, 85]) сходится к решению системы (II.52) со скоростью геометрической прогрессии, знаменатель которой

$$\rho = \frac{\Delta - \delta}{\Delta + \delta}. \quad (\text{II.80})$$

Вместо итерационного процесса (II.53) для ускорения сходимости итераций используют процесс, который в канонической форме может быть записан в виде

$$B \frac{y^{(k+1)} - y^{(k)}}{\tau_{k+1}} + Ay^{(k)} = f, \quad y^{(0)} = y_0, \quad k = 0, 1, \dots \quad (\text{II.81})$$

Итерационный параметр τ_i в этом случае вычисляется по формуле

$$\tau_i = \frac{2}{(\gamma_2 + \gamma_1) - (\gamma_2 - \gamma_1) \cos \frac{(2i-1)\pi}{2s}}, \quad i = 1, 2, \dots, s, \quad (\text{II.82})$$

где s — число шагов в цикле.

Для численной устойчивости счета необходимо, чтобы выбор параметров каждого цикла осуществлялся в определенной последовательности [58, 85]. Приведем одну из возможных очередностей $\sigma^{(s)} = (\sigma_1, \sigma_2, \dots, \sigma_s)$ для последовательности параметров τ_i ($i = 1, 2, \dots, s$), когда s является степенью числа 2, т. е. $s = 2^r$.

При $r = 1$ положим $\sigma^{(2)} = (1, 2)$. Пусть при $s = 2^{r-1}$ порядок $\sigma^{(2^{r-1})}$ установлен, т. е.

$$\sigma^{(2^{r-1})} = (\sigma_1, \sigma_2, \dots, \sigma_{2^{r-1}}),$$

тогда для $\sigma^{(2^r)}$ полагаем

$$\sigma^{(2^r)} = (\sigma_1, 2^r + 1 - \sigma_1, \sigma_2, 2^r + 1 - \sigma_2, \dots, 2^r + 1 - \sigma_{2^{r-1}}).$$

Таким образом, для определения $\sigma^{(s)}$ приведены рекуррентные формулы. В частности, при $r = 2, 3, 4$ соответственно получим (1, 4, 2, 3); (1, 8, 4, 5, 2, 7, 3, 6); (1, 16, 8, 9, 4, 13, 5, 12, 2, 15, 7, 10, 3, 14, 6, 11).

Итерационный процесс (II.81) с выбором параметров по формуле (II.82) можно заканчивать при выполнении условия (II.40) или условия

$$\frac{\|y^{(k+1)} - y^{(k)}\|}{\|y^{(k)}\|} \leq \frac{\varepsilon}{1 + \varepsilon} \frac{\min |\tau_{k+1}|}{\|A^{-1}\|}.$$

5. Ускорение сходимости итераций. Итерационные процессы, записываемые в канонической форме (II.53), исследованы при произвольных симметричных и положительно определенных матрицах B .

От конкретного вида B зависят как число арифметических операций, так и скорость сходимости итераций. Так как скорость сходимости зависит от ρ , то естественно строить B и выбирать итерационные параметры исходя из конкретного значения A таким образом, чтобы как величина ρ , так и число арифметических операций по возможности были минимальными. Матрица B может быть выписана в явном виде и построена в ходе итерационного процесса.

Если матрица A , для которой известны оценки снизу минимального δ и сверху максимального Δ собственных значений, представима в виде суммы более простых симметричных, положительно определенных перестановочных между собой матриц A_α , $\alpha = 1, \dots, p$, т. е.

$$A = \sum_{\alpha=1}^p A_\alpha, \quad A_\alpha = A_\alpha^T, \quad (A_\alpha y, y) \geq c \|y\|^2, \quad c > 0, \quad (\text{II.83})$$

$$\frac{\delta}{p} \|y\|^2 \leq (A_\alpha y, y) \leq \frac{\Delta}{p} \|y\|^2, \quad \alpha = 1, 2, \dots, p, \quad (\text{II.84})$$

то в этом случае матрицу B в канонической форме (II.53) целесообразно определять по формуле

$$B = \prod_{\alpha=1}^p (E + \omega A_\alpha).$$

При этом ρ для формулы (II.53) достигает минимального значения при

$$\omega = \frac{p(\sqrt[p]{\Delta} - \sqrt[p]{\delta})}{\sqrt[p]{\delta\Delta}(\sqrt[p]{\Delta^{p-1}} - \sqrt[p]{\delta^{p-1}})}, \quad \tau = \frac{2}{\gamma_1 + \gamma_1} \quad (\text{II.85})$$

и вычисляется по формуле

$$\rho = \frac{((p-1)\delta + \Delta)\left(1 + \omega \frac{\delta}{p}\right) - \delta p \left(1 - \omega \frac{\Delta}{p}\right)}{((p-1)\delta + \Delta)\left(1 + \omega \frac{\delta}{p}\right) + \delta p \left(1 - \omega \frac{\Delta}{p}\right)}, \quad (\text{II.86})$$

где

$$\gamma_1 = \frac{\delta}{\left(1 + \omega \frac{\delta}{p}\right)^p}, \quad \gamma_2 = \frac{(p-1)\delta + \Delta}{p \left(1 + \omega \frac{\Delta}{p}\right) \left(1 + \omega \frac{\delta}{p}\right)^{p-1}}. \quad (\text{II.87})$$

Одношаговый итерационный процесс при таком выборе матрицы B сводится к решению для каждого $k = 0, 1, 2, \dots$ (k — номер итерации) последовательности задач

$$\begin{aligned} (E + \omega A_1) y^{(k + \frac{1}{p+1})} &= r^{(k)}, \\ (E + \omega A_2) y^{(k + \frac{2}{p+1})} &= y^{(k + \frac{1}{p+1})}, \\ &\dots \end{aligned}$$

$$(E + \omega A_p) y^{(k + \frac{p}{p+1})} = y^{(k + \frac{p-1}{p+1})},$$

$$y^{(k+1)} = y^{(k)} + \tau y^{(k + \frac{p}{p+1})},$$

где $r^{(k)} = f - Ay^{(k)}$ — невязка на k -й итерации.

Из рассмотрения знаменателей геометрической прогрессии (II.86) при $p = 2$ и (II.80) следует очевидное преимущество итерационных схем с матрицей B .

II.3. ОДНОШАГОВЫЕ РЕЛАКСАЦИОННЫЕ МЕТОДЫ

1. Метод Гаусса — Зейделя. Для решения системы (II.34) можно применить еще один класс итерационных процессов — релаксационные методы. Простейшим из них является метод Гаусса — Зейделя. Этот метод в матричном виде можно записать следующим образом:

$$x^{(k+1)} = -D^{-1}Lx^{(k+1)} - D^{-1}Rx^{(k)} + D^{-1}f, \quad (II.88)$$

где

$$L = \begin{bmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ a_{31} & a_{32} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix}, \quad D = \begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & a_{nn} \end{bmatrix},$$

$$R = \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & 0 & a_{23} & \dots & a_{2n} \\ 0 & 0 & 0 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Заметим, что метод Гаусса — Зейделя в канонической форме (II.48) имеет вид

$$(L + D)(x^{(k+1)} - x^{(k)}) + Ax^{(k)} = f, \quad x^{(0)} = x_0 \quad (II.89)$$

и может быть представлен так:

$$(L + D)x^{(k+1)} + Rx^{(k)} = f, \quad x^{(0)} = x_0,$$

или

$$x^{(k+1)} = -(L + D)^{-1}Rx^{(k)} + (L + D)^{-1}f. \quad (II.90)$$

Для сходимости этого метода при произвольном выборе начального приближения $x^{(0)} = x_0$ и любом значении f необходимо и достаточно,

чтобы корни уравнения

$$\begin{vmatrix} a_{11}\lambda & a_{12} & \dots & a_{1n} \\ a_{21}\lambda & a_{22}\lambda & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}\lambda & a_{n2}\lambda & \dots & a_{nn}\lambda \end{vmatrix} = 0 \quad (II.91)$$

были по модулю меньше единицы.

Действительно, из представления метода Гаусса — Зейделя в форме (II.90) следует, что для его сходимости необходимо и достаточно, чтобы все собственные числа матрицы $M = -(L + D)^{-1}R$ были по модулю меньше единицы. Таким образом приходим к задаче

$$|M - \lambda E| = |-(L + D)^{-1}R - \lambda E| = 0.$$

Умножив обе части последнего равенства на $|-(L + D)|$, получаем

$$|R + (L + D)\lambda| = 0,$$

откуда следует

$$|R + (L + D)\lambda| = 0$$

или в развернутой форме уравнение (II.91).

Достаточные условия применимости метода Гаусса — Зейделя при другом определении матрицы M такие же, как и метода Якоби. Области сходимости методов Якоби и Гаусса — Зейделя различны и лишь частично перекрываются. Иногда метод Гаусса — Зейделя сходится быстрее, чем метод Якоби. Однако он не всегда оказывается более выгодным по сравнению с методом Якоби.

Если матрица A системы линейных алгебраических уравнений (II.34) симметрична и невырождена и если все диагональные элементы положительны, то метод Гаусса — Зейделя сходится тогда и только тогда, когда A — положительно определенная матрица.

Заканчивать итерационный процесс можно, проверяя условие (II.40) или условие

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \leq \frac{\varepsilon}{1 + \varepsilon \|A^{-1}\| \|L + D\|}, \quad (II.92)$$

что гарантирует выполнение неравенства (II.41). Покажем это для условия (II.92). Из (II.89) можно записать

$$(L + D)(x^{(k+1)} - x^{(k)}) = A(x^* - x^{(k)}),$$

и, следовательно,

$$\|x^{(k)} - x^*\| \leq \|A^{-1}\| \|L + D\| \|x^{(k+1)} - x^{(k)}\|.$$

Если $\|x^{(k)}\| \neq 0$, то можно записать

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \geq \frac{\|x^{(k)} - x^*\|}{\|A^{-1}\| \|L + D\| \|x^{(k)}\|},$$

$$\frac{\|x^{(k)} - x^*\|}{\|x^{(k)}\|} \leq \frac{\varepsilon}{1 + \varepsilon}.$$

Таблица 3

$x^{(k)}$	k					
	0	1	2	3	4	5
$x_1^{(k)}$	0	0,750 000	0,870 833	0,997 708	1,002 03	1,000 14
$x_2^{(k)}$	0	1,450 000	2,005 840	2,009 46	2,000 38	1,999 86
$x_3^{(k)}$	0	2,900 000	3,045 000	3,003 90	2,999 45	2,999 90

Далее, можно получить

$$\|x^{(k)} - x^*\| + \varepsilon \|x^{(k)} - x^*\| \leq \varepsilon \|x^{(k)}\| = \varepsilon \|x^{(k)} - x^* + x^*\| \leq \varepsilon \|x^{(k)} - x^*\| + \varepsilon \|x^*\|,$$

откуда следует неравенство $\|x^{(k)} - x^*\| \leq \varepsilon \|x^*\|$ и, наконец, неравенство (II.41), что и требовалось доказать [67].

Вычислительная схема этого метода может быть реализована следующим образом. До начала итерационного процесса задается относительная погрешность ε , с которой нужно получить решение задачи, и начальное приближение к решению $x^{(0)} = x_0$. Затем вычисления осуществляем по формулам

$$x_i^{(k+1)} = \left(f_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) / a_{ii} \quad (\text{II.93})$$

и после каждой итерации проверяем условие (II.40) или (II.92). На одну итерацию в методе Гаусса — Зейделя требуются n делений, $n^2 - n$ умножений и n^2 сложений.

Пример 2. Методом Гаусса — Зейделя решить систему линейных алгебраических уравнений (II.46) с относительной погрешностью $\varepsilon = 10^{-2}$.

Решение. Вначале убеждаемся, что достаточные условия применимости метода выполнены, после чего, задаваясь нулевым начальным приближением, вычисляем $x_i^{(k+1)}$ по формулам (II.93) и, проверяя на каждой итерации условие окончания (II.92), получаем приближение к решению системы уравнений (II.46), приведенное в табл. 3.

Заметим, что неравенство (II.40) для $\varepsilon = 10^{-2}$ было выполнено за четыре итерации. Ответ совпадает с приведенным в табл. 3 для случая $k = 4$.

Отметим, что метод Гаусса — Зейделя обладает медленной сходимостью, и поэтому его, как и метод Якоби, нецелесообразно применять для решения систем линейных алгебраических уравнений высокого порядка и плохо обусловленных. При использовании релаксационных методов для частных случаев матриц, например симметричных, можно строить итерационные процессы, обладающие более быстрой сходимостью, чем метод Гаусса — Зейделя. Один из таких методов рассмотрен ниже.

2. Метод верхней релаксации. Если матрица системы линейных алгебраических уравнений (II.34) симметрична и положительно определена, то для ее решения можно использовать метод верхней релакса-

ции, который обладает более высокой скоростью сходимости по сравнению с методом Гаусса — Зейделя. В этом методе, исходя из k -го приближения к решению, вычисляют

$$x^{(k+1/2)} = -D^{-1} L x^{(k-1)} - D^{-1} R x^{(k)} + D^{-1} f, \quad (\text{II.94})$$

а затем $(k+1)$ -е приближение к решению находят по формуле

$$x^{(k+1)} = x^{(k)} + \tau (x^{(k+1/2)} - x^{(k)}), \quad (\text{II.95})$$

где τ — параметр, ускоряющий сходимость итераций. Итерационный процесс по схеме (II.94), (II.95) в канонической форме (II.48) может быть записан следующим образом:

$$\left(L + \frac{1}{\tau} D \right) (x^{(k+1)} - x^{(k)}) + A x^{(k)} = f, \quad x^{(0)} = x_0. \quad (\text{II.96})$$

Итерационный процесс, записанный в форме (II.96), сходится при $0 < \tau < 2$, если матрица системы линейных алгебраических уравнений симметрична и положительно определена.

При $1 < \tau < 2$ рассматриваемый итерационный процесс называют методом верхней релаксации, при $0 < \tau < 1$ — методом нижней релаксации, при $\tau = 1$ получают метод Гаусса — Зейделя.

Итерационный процесс (II.94), (II.95) целесообразно заканчивать при выполнении условия (II.40) или условия

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \leq \frac{\varepsilon}{1 + \varepsilon} \frac{1}{\|A^{-1}\| \|L + \frac{1}{\tau} D\|}, \quad (\text{II.97})$$

которое гарантирует выполнение условия (II.41); в нем ε — относительная погрешность получаемого решения.

Выбор оптимального параметра τ в общем случае — задача трудная. Существуют различные модификации метода верхней релаксации, отличающиеся количеством одновременно уточняющихся компонент и выбором итерационного параметра τ [61, 150, 192]. Метод верхней релаксации является одним из эффективных одношаговых итерационных методов.

Пример 3. Методом верхней релаксации решить систему линейных алгебраических уравнений

$$\begin{aligned} 2x_1 - x_2 &= 0, \\ -x_1 + 2x_2 - x_3 &= 0, \\ -x_2 + 2x_3 &= 4, \end{aligned} \quad (\text{II.98})$$

используя в качестве итерационного параметра $\tau = 1,171\,577$ и условие (II.97) для $\varepsilon = 10^{-2}$.

Решение. Так как матрица системы линейных алгебраических уравнений (II.98) симметрична и положительно определена, то для ее решения можно применять метод верхней релаксации. Для решения этой системы использовалась электронная вычислительная машина МИР-2 с длиной машинного слова в шесть десятичных знаков. Приближенное решение системы (II.98), полученное методом верхней релаксации с усло-

вием окончания итераций (II.97) за пять итераций, имеет вид

$$x^{(5)} = \begin{bmatrix} 0,998\ 23 \\ 1,998\ 94 \\ 2,999\ 65 \end{bmatrix},$$

в то время как математическое решение системы (II.98)

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Отметим, что решение той же системы методом Гаусса — Зейделя с условием окончания итераций (II.92) было получено за тринадцать итераций:

$$x^{(13)} = \begin{bmatrix} 0,999\ 51 \\ 1,999\ 51 \\ 2,999\ 75 \end{bmatrix},$$

а методом Якоби с условием окончания итераций (II.43) — за двадцать четыре итерации

$$x^{(24)} = \begin{bmatrix} 0,999\ 51 \\ 1,999\ 51 \\ 2,999\ 51 \end{bmatrix}.$$

II.4. НЕКОТОРЫЕ ВОПРОСЫ МАШИННОЙ РЕАЛИЗАЦИИ ОДНОШАГОВЫХ ИТЕРАЦИОННЫХ ПРОЦЕССОВ

1. Примеры теоретически сходящихся, но не реализуемых на ЭВМ процессов. До сих пор нами предполагалось, что при реализации итерационных процессов на ЭВМ числа, с которыми оперируют машины, не имеют ограничений. Однако это не так. В результате может возникнуть ситуация, при которой теоретически сходящийся итерационный процесс при его реализации на машине будет давать машинное решение, отличное от математического решения задачи.

Так, использование для решения системы

$$Ax = b, \quad (II.99)$$

где

$$A = \begin{bmatrix} 10 & 0,01 & 0 \\ 0,01 & 0,001 & 0,1 \\ 0 & 0,1 & 1000 \end{bmatrix}, \quad b = \begin{bmatrix} 20,05 \\ 0,125 \\ 1000,5 \end{bmatrix}, \quad (II.100)$$

с симметричной и положительно определенной матрицей⁸ A явного одношагового итерационного процесса (II.78) с выбором итерационного параметра τ по формуле (II.79) ($\tau = 0,001\ 999\ 998\ 06$) теоретически вполне оправдано, так как выполнены все необходимые и достаточные условия для сходимости итераций со скоростью геометрической прогрессии со знаменателем $\rho = 0,999\ 998\ 08$, вычисляемым по формуле (II.80).

Реализация этого итерационного процесса на ЕС 1020 (машинное слово имеет 32 бита, длина мантиссы машинного слова — 24 бита)

⁸ Матрица A имеет собственные значения $\lambda_1 = 0,000\ 979\ 999$, $\lambda_2 = 10,000\ 01$, $\lambda_3 = 1000,000\ 01$.

с условием окончания итераций

$$\max_i \left| \frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k)}} \right| \leq \frac{\tau \delta \epsilon}{1 + \epsilon}, \quad \epsilon = 10^{-3} \quad (II.101)$$

по ФОРТРАН-программе для нулевого начального приближения дает на нечетной итерации $\bar{x}$, а на четной итерации $\bar{x}$, где

$$\bar{x} = \begin{bmatrix} 2,000\ 455\ 9 \\ 4,546\ 527\ 9 \\ 0,952\ 036\ 86 \end{bmatrix}, \quad \bar{x} = \begin{bmatrix} 2,000\ 455\ 9 \\ 4,546\ 537\ 4 \\ 1,048\ 033\ 7 \end{bmatrix},$$

в то время как математическое решение задачи

$$x = \begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix}.$$

Использование в той же ФОРТРАН-программе двойного машинного слова позволяет получить за 4 000 000 итераций машинное решение

$$x^{(400\ 000\ 0)} = \begin{bmatrix} 2,000\ 001968 \\ 4,998\ 032\ 428 \\ 0,999\ 606\ 327\ 7 \end{bmatrix},$$

достаточно близкое к математическому решению системы (II.99), (II.100).

Теперь рассмотрим систему линейных алгебраических уравнений (II.99), матрица и правая часть которой имеют следующие значения:

$$A = \begin{bmatrix} 1,416 & 1 & 0 \\ 1 & 1,416 & 1 \\ 0 & 1 & 1,416 \end{bmatrix}, \quad b = \begin{bmatrix} 7,832 \\ 10,08 \\ 6,416 \end{bmatrix}. \quad (II.102)$$

Исследуем возможность решения системы (II.99), (II.102) методом Якоби (см. п. 2 параграфа II.2). Так как максимальное по модулю собственное значение итерируемой матрицы M в (II.35) меньше единицы ($|\mu| = 0,9985$), то выполнены необходимые и достаточные условия для сходимости метода, реализуемого по формулам (II.36). Тем не менее реализация этого метода с нулевым начальным приближением, с условием окончания процесса (II.43) и $\epsilon = 10^{-2}$ на ЭВМ МИР-1 с длиной машинного слова в четыре десятичных знака дает

$$x^{(100\ 0)} = \begin{bmatrix} 0,509\ 1 \\ 4,632 \\ -0,490\ 8 \end{bmatrix}, \quad x^{(1001)} = \begin{bmatrix} 2,260 \\ 7,111 \\ 1,260 \end{bmatrix}.$$

Реализация той же программы на машине МИР-1 с длиной машинного слова десять десятичных знаков дает

$$x^{(956\ 6)} = \begin{bmatrix} 1,999\ 995\ 272 \\ 4,999\ 979\ 618 \\ 0,999\ 995\ 72 \end{bmatrix}.$$

$$x = \begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix}.$$

Эти примеры показывают, что теоретическое исследование итерационных процессов без учета машинной реализации является лишь необходимым, но не достаточным условием успешного применения их на практике.

В принципе отличие теоретического итерационного процесса от реально реализуемого определяется рядом факторов. Отметим некоторые из них. Вместо исходной матрицы и правой части системы в ЭВМ имеют дело с некоторой возмущенной матрицей и правой частью (см., например, п. 5 параграфа I.2). Свойства этой другой системы по существу не исследовались. Если исходная система обладает малым, но отличным от нуля определителем и для нее были выполнены условия сходимости итерационного процесса, то машинная матрица может не обладать этими свойствами.

2. Некоторые вопросы достоверности решений, получаемых итерационными методами. Как следует из сказанного выше, итерационные процессы дают лишь приближенное решение системы линейных алгебраических уравнений, поэтому машинное решение системы будет отличаться от математического решения ее. Исследование системы линейных алгебраических уравнений, находящейся в ЭВМ, которое можно провести с помощью самой машины, позволяет правильно установить специфику системы, выбрать соответствующий метод решения и в некоторых случаях оценивать влияние погрешности машинного перевода чисел из одной системы счисления в другую.

Сходящийся итерационный процесс теоретически должен давать точное решение системы, когда номер итерации $k \rightarrow \infty$. На практике условия окончания итерационного процесса зависят от свойств матрицы системы, метода решения, желаемой точности решения и длины машинного слова (см. условия окончания итерационных процессов в п. 2 и 4 параграфа II.2, а также п. 1 и 2 параграфа II.3).

Отметим, что компоненты $x^{(k)}$, которые участвуют во всех указанных выше условиях, включают в себя все погрешности вычислений, вносимые способом построения компонент.

Выбор ε (относительной погрешности получаемого машинного решения) в условиях окончания итерационных процессов определяется несколькими факторами. Одним из них является требуемая точность получаемого решения. При этом надо учитывать, что значение ε не должно быть меньше погрешности машинной реализации и правая часть условий окончания итераций (II.40), (II.43), (II.67), (II.92), (II.97) должна быть больше машинного нуля.

Реализация на ЭВМ ЕС 1020 итерационной процедуры (II.78), (II.79) для решения системы (II.99), (II.100) с двойным машинным словом позволила получить решение рассматриваемой задачи с заданной точностью за $4 \cdot 10^6$ итераций. Уменьшению машинной погрешности

будет служить вычисление скалярного произведения по методу накопления.

После определения ε целесообразно определить число арифметических операций, необходимых для получения решения с заданной точностью. Если это число окажется достаточно большим и неприемлемым для реализации на машине, то иногда перед реализацией итерационного процесса исходную систему можно преобразовать с помощью эквивалентных преобразований так, чтобы уменьшить число обусловленности системы.

Так, (II.99), (II.100) путем эквивалентных преобразований можно привести к системе

$$\begin{aligned} 10x_1 + x_2 &= 25, \\ x_1 + 10x_2 + x_3 &= 53, \\ x_2 + 10x_3 &= 15 \end{aligned} \quad (\text{II.103})$$

с симметричной и положительно определенной матрицей, собственные значения которой следующие:

$$\lambda_1 = 8,585\,786\,4, \quad \lambda_2 = 10, \quad \lambda_3 = 11,414\,214.$$

Использование для решения системы (II.103) той же ФОРТРАН-программы, которая использовалась для решения системы (II.99), (II.100), реализация этой программы на ЭВМ ЕС 1020 с одинарным машинным словом и условием окончания итерации (II.101) для $\varepsilon = 10^{-3}$ позволяет получить машинное решение

$$x^{(1)} = \begin{bmatrix} 2,000\,200\,3 \\ 5,000\,119\,2 \\ 1,000\,199\,3 \end{bmatrix},$$

мало отличающееся от математического решения, всего за 5 итераций.

Если затруднительно минимизировать число обусловленности матрицы системы путем эквивалентных преобразований, то применение другого итерационного процесса может дать неплохие результаты.

Так, применение для решения системы (II.99), (II.100) итерационного метода, реализуемого по формуле (II.36) на ЭВМ ЕС 1020 с одинарным машинным словом и условием окончания итерации (II.40) для $\varepsilon = 10^{-3}$, позволяет получить машинное решение

$$x^{(1)} = \begin{bmatrix} 2,000\,000\,0 \\ 5,000\,066\,8 \\ 1,000\,000\,0 \end{bmatrix}$$

всего за три итерации. Переход на другой метод особенно целесообразен, если он обладает более быстрой по сравнению с предыдущим методом скоростью сходимости.

Если при оценке числа арифметических операций видно, что задача будет решаться слишком долго на данной машине, то целесообразно перейти к решению на более быстродействующей электронной вычислительной машине.

В условия окончания итерационного процесса, гарантирующие достижение заданной точности, входят $\text{cond } A$ или норма обратной матрицы $\|A^{-1}\|$ ⁹. При решении некоторых прикладных задач с симметричными и положительно определенными матрицами в ряде случаев можно дать оценку снизу минимального и сверху максимального собственных значений, а следовательно, числа обусловленности матриц (см., например, [66]). Зная оценку числа обусловленности, ее можно подставить в условия окончания итерационного процесса исходной задачи и получить тем самым машинное решение, достаточно близкое к математическому решению задачи. Используя оценку числа обусловленности матрицы и погрешность в задании исходных данных ΔA , Δb , можно оценить по формулам (I.14), (I.15) близость математического и физического решений задачи. (Некоторые вопросы машинной реализации итерационных процессов рассмотрены в работе [170].)

Для решения систем линейных алгебраических уравнений высокого порядка с одной или небольшим числом правых частей целесообразно использовать итерационные методы. Они просты в реализации на ЭВМ. В ряде случаев¹⁰ итерационные методы позволяют по сравнению с прямыми методами существенно экономить память ЭВМ. Кроме того, в этих методах заложена возможность получения решения задачи заданной точности с меньшим числом арифметических операций. Эта возможность может быть реализована при использовании так называемых быстросходящихся итерационных методов.

Однако каждый итерационный процесс характеризуется условиями сходимости, практическая проверка которых должна предшествовать применению этого метода. Для успешной реализации итерационных методов важна быстрота сходимости получаемых приближений к точному решению системы. Построение таких методов требует дополнительной информации о матрице системы, которую в общем случае получить трудно.

При решении вопроса о применении к системе линейных алгебраических уравнений прямого или итерационного метода необходимо учитывать специфику матрицы системы, возможность получения оценок спектра и сжатия его, требуемые объемы памяти для прямых и итерационных методов, возможность задания исходной информации для решения системы в виде Ay , требуемое количество арифметических операций для получения решения задачи с заданной степенью точности, а также технические и математические особенности имеющейся ЭВМ (ее быстродействие, объемы памяти различных ступеней, длина

машинного слова, режим фиксированной или плавающей точки, процедура округления, возможность выполнения вычислений с удвоенной точностью и т. д.).

Сопоставление объема требуемой памяти, числа арифметических операций и времени решения задачи на конкретной машине позволяет сделать вывод о целесообразности применения к конкретной системе линейных алгебраических уравнений прямого или итерационного метода решения.

Для решения систем линейных алгебраических уравнений с несимметричными матрицами иногда можно использовать метод Якоби. Перед применением метода Якоби при невыполнении достаточных условий сходимости иногда полезно проверить необходимые и достаточные условия сходимости, вычислив максимальное собственное значение итерируемой матрицы M степенным методом (см. п. 2 параграфа III.4).

Если позволяют память ЭВМ и длина машинного слова, то для решения системы линейных алгебраических уравнений (II.34) с несимметричной матрицей можно использовать так называемые двухшаговые методы. В двухшаговых итерационных процессах $(k+1)$ -е приближение зависит от k - и $(k-1)$ -го приближений. Одним из двухшаговых итерационных методов является метод сопряженных градиентов, реализуемый по формулам [115]

$$x^{(0)} = 0, \quad r^{(0)} = f, \quad s^{(-1)} = 0, \quad (p^{(-1)}, p^{(-1)}) = 1, \quad p^{(k)} = A^T r^{(k)},$$

$$b^{(k-1)} = \frac{(p^{(k)}, p^{(k)})}{(p^{(k-1)}, p^{(k-1)})}, \quad s^{(k)} = p^{(k)} + b^{(k-1)} s^{(k-1)}, \quad \xi^{(k)} = A s^{(k)},$$

$$a^{(k)} = \frac{(p^{(k)}, p^{(k)})}{(\xi^{(k)}, \xi^{(k)})}, \quad x^{(k+1)} = x^{(k)} + a^{(k)} s^{(k)}, \quad r^{(k+1)} = r^{(k)} - a^{(k)} \xi^{(k)},$$

$$k = 0, 1, 2, \dots$$

Условием окончания процесса итераций может служить

$$\frac{\|A^T f - A^T A x\|}{\|A^T f\|} \leq \frac{\varepsilon}{\text{cond } A^T A}.$$

Отметим, что этот итерационный процесс осуществляет неявно решение системы линейных алгебраических уравнений $A^T A x = A^T f$, поэтому требует для своей реализации достаточной длины машинного слова.

Метод сопряженных градиентов в данной реализации всегда будет сходящимся, в то время как метод простой итерации может быть в зависимости от свойств системы сходящимся или расходящимся.

Наиболее изученными являются итерационные процессы решения систем линейных алгебраических уравнений с симметричными положительно определенными матрицами.

Системы линейных алгебраических уравнений с симметричными и положительно определенными матрицами целесообразно решать по явным или неявным одношаговым итерационным схемам с переменным набором параметров, обеспечивающим численную устойчивость

⁹ Зная число обусловленности матрицы, легко вычислить

$$A^{-1} = \frac{\text{cond } A}{\|A\|}.$$

¹⁰ К таким случаям можно отнести решение систем с порождающимися матрицами и запись систем с матрицами, обладающими определенной спецификой, в виде процедуры вычисления вектора Ay , если эта процедура требует меньшей памяти, чем хранение разреженной матрицы и вектора правых частей, необходимых при реализации прямых методов.

итерационных процессов, или методом верхней релаксации. Если в методе верхней релаксации теоретически трудно выбрать параметр релаксации, то в ряде случаев его определяют экспериментально. Если для заданной матрицы A известны энергетически эквивалентная ей матрица B и оценки констант энергетической эквивалентности, то, используя матрицу B , можно ускорить сходимость явных итерационных процессов.

Когда затруднено получение отличной от нулевой оценки снизу для минимального собственного значения, то иногда можно использовать для решения системы метод наискорейшего спуска [37], реализуемый по формулам

$$y^{(k+1)} = y^{(k)} + \alpha_k r^{(k)}, \quad r^{(k)} = f - Ay^{(k)},$$

$$\alpha_k = \frac{(r^{(k)}, r^{(k)})}{(Ar^{(k)}, r^{(k)})}, \quad k = 0, 1, 2, \dots$$

и минимизирующий функцию ошибки, или метод минимальных невязок [46]

$$y^{(k+1)} = y^{(k)} + \alpha_k r^{(k)}, \quad r^{(k)} = f - Ay^{(k)},$$

$$\alpha_k = \frac{(Ar^{(k)}, r^{(k)})}{(Ar^{(k)}, Ar^{(k)})}, \quad k = 0, 1, 2, \dots,$$

минимизирующий вектор невязки.

Если для матрицы A можно построить энергетически эквивалентную ей и легко обратимую прямым или итерационным методом матрицу B , а константы энергетической эквивалентности γ_1, γ_2 или неизвестны, или вычисляются слишком грубо, то для решения системы (II.52) можно использовать двухступенчатый метод, в котором для вычисления $\{\alpha_k\}$ можно использовать либо формулу метода наискорейшего спуска, либо формулу метода минимальных поправок [85].

Вычислительная схема этого метода может быть реализована следующим образом. Сначала одним из прямых или быстросходящихся итерационных методов решается вспомогательная задача относительно поправки

$$B\omega^{(k)} = r^{(k)}, \quad r^{(k)} = f - Ay^{(k)},$$

затем в случае неявного метода наискорейшего спуска вычисляется

$$\alpha_{k+1} = \frac{(r^{(k)}, \omega^{(k)})}{(A\omega^{(k)}, \omega^{(k)})},$$

или в случае неявного метода минимальных поправок —

$$\alpha_{k+1} = \frac{(\omega^{(k)}, A\omega^{(k)})}{(B^{-1}A\omega^{(k)}, A\omega^{(k)})}.$$

Приближение на $(k+1)$ -й итерации вычисляют по формуле

$$y^{(k+1)} = y^{(k)} + \alpha_{k+1} \omega^{(k)}.$$

Отметим, что двухступенчатые итерационные методы могут быть построены на основе других как одношаговых, так и двухшаговых итерационных процессов (см., например, [66, 85]).

Если для положительно определенной матрицы A системы линейных алгебраических уравнений (II.52) можно построить энергетически эквивалентную и легко обратимую каким-либо прямым методом матрицу B и в неравенствах

$$\gamma_1 (By, y) \leq (Ay, y) \leq \gamma_2 (By, y)$$

можно получить константы эквивалентности γ_1 и γ_2 , то из одношагового итерационного процесса, записанного в канонической форме

$$B \frac{y^{(k+1)} - y^{(k)}}{\tau_{k+1}} + Ay^{(k)} = f, \quad y^{(0)} = y_0,$$

можно построить двухступенчатый итерационный процесс.

Обозначая $\omega^{(k)} = \frac{y^{(k+1)} - y^{(k)}}{\tau_{k+1}}$, вначале решают вспомогательную задачу относительно поправки

$$B\omega^{(k)} = r^{(k)}, \quad r^{(k)} = f - Ay^{(k)}$$

прямым или быстросходящимся итерационным методом, а затем, учитывая формулу для поправки $\omega^{(k)}$, находят следующее приближение на $(k+1)$ -й итерации по формуле

$$y^{(k+1)} = y^{(k)} + \tau_{k+1} \omega^{(k)},$$

где $\{\tau_k\}$ — упорядоченная последовательность оптимального набора итерационных параметров (см. п. 4 параграфа II.2).

Для решения систем линейных алгебраических уравнений с симметричными положительно полуопределенными матрицами высокого порядка целесообразно использовать итерационные методы, рассмотренные в работе [65]. Итерационным методам решения систем линейных алгебраических уравнений, в том числе и с несимметричными матрицами, посвящены работы [54, 166].

Для решения систем линейных алгебраических уравнений с симметричными матрицами специального вида, получаемыми, например, методом конечных разностей или конечных элементов исходя из общей теории итерационных процессов, необходимо строить специальные итерационные методы, позволяющие в максимальной степени учитывать вид и свойства этих матриц и за счет этого получить высокую скорость сходимости итераций (см., например, [61, 66, 87, 192] и др.).

Если позволяет память ЭВМ, то вместо одношаговых итерационных процессов для решения систем линейных алгебраических уравнений с симметричными и положительно определенными матрицами целесообразно использовать двухшаговые итерационные процессы.

Некоторые двухшаговые итерационные процессы в канонической форме А. А. Самарского [85] можно записать в виде

$$B \left[\frac{y^{(k+1)} - y^{(k-1)}}{2\tau} + \kappa (y^{(k+1)} - 2y^{(k)} + y^{(k-1)}) \right] + Ay^{(k)} = f,$$

$$y^{(0)} = y_0, \quad y^{(1)} = y_1,$$

где B — положительно определенная матрица, τ и κ параметры, обеспечивающие сходимость итерационного процесса, $\tau = \frac{1}{\sqrt{\gamma_1 \gamma_2}}$, $\kappa = \frac{\gamma_1 + \gamma_2}{4}$, причем

$$\gamma_1 (By, y) \leq (Ay, y) \leq \gamma_2 (By, y).$$

Вводя поправку $\omega^{(k)} = \frac{y^{(k+1)} - y^{(k-1)}}{2\tau} + \kappa (y^{(k+1)} - 2y^{(k)} + y^{(k-1)})$,

можно получить двухступенчатый двухшаговый итерационный процесс. Сначала решаем вспомогательную задачу относительно поправки $\omega^{(k)}$

$$B\omega^{(k)} = r^{(k)}, \quad r^{(k)} = f - Ay^{(k)}$$

прямым или быстроходящимся итерационным методом, а затем, используя формулы для поправки, вычисляем $(k+1)$ -е приближение по формуле

$$y^{(k+1)} = \frac{2\tau}{1+2\tau\kappa} (\omega^{(k)} + 2\kappa y^{(k)}) + \frac{1-2\tau\kappa}{1+2\tau\kappa} y^{(k-1)}.$$

Представителем двухшаговых методов решения систем линейных алгебраических уравнений с симметричными и положительно определенными матрицами является метод сопряженных градиентов [98, 101, 160].

Перед применением метода сопряженных градиентов исходную систему линейных алгебраических уравнений с симметричной и положительно определенной матрицей целесообразно преобразовать так, чтобы она осталась симметричной, а ее диагональные элементы оказались равными единице.

Преобразование исходной системы линейных алгебраических уравнений в эквивалентную ей систему в целях минимизации числа обусловленности матрицы системы является весьма полезным как для одношаговых, так и для двухшаговых итерационных процессов. (О таких преобразованиях см., например, в работах [147, 155].)

Исследование итерационных процессов с точки зрения их машинной реализации позволяет решить вопрос о близости машинного и математического решений задачи. Влияние ошибок машинной реализации различных итерационных процессов решения систем линейных алгебраических уравнений исследовалось в ряде работ (см., например, [39—42] и др.).

Вопросы близости математического и физического решений задач могут быть решены так же, как это сделано в гл. I. В случае, когда для симметричной матрицы A можно получить оценку снизу наименьшего δ , оценку сверху наибольшего Δ собственных значений, число обусловленности матрицы ($\text{cond } A$) может быть оценено сверху значением Δ/δ , а число обусловленности системы линейных алгебраических уравнений

$$m \leq \frac{\Delta}{\delta} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta f\|}{\|f\|} \right),$$

где ΔA и Δf — погрешности в задании элементов матрицы и правой части системы.

Для оценки достоверности получаемого итерационным методом решения систем линейных уравнений иногда используют интервальную арифметику (см., например, [117, 123, 167] и др.).

При выборе итерационного метода решения задачи обычно исходят из свойств матрицы системы линейных алгебраических уравнений (несимметричная или симметричная, положительно определенная, положительно полуопределенная, незнакоопределенная, разреженная или плотная, порядок ее и т. д.), возможности получения оценок спектра и построения некоторых вспомогательных матриц, энергетически эквивалентных исходным матрицам систем, возможности проведения эквивалентных преобразований над исходной системой с целью минимизации числа обусловленности. Выбор итерационного метода диктуется также техническими и математическими особенностями конкретной ЭВМ и ее операционной системы (длины машинного слова, быстродействия, объемов запоминающих устройств различного уровня, режима фиксированной или плавающей запятой, возможности аппаратного или программного вычисления с удвоенной точностью и т. д.).

Отметим, что исследования в области решения систем линейных алгебраических уравнений итерационными методами в последние годы сосредоточивались вокруг построения итерационных методов решения, обладающих за счет учета специфики матрицы системы высокими скоростями сходимости итераций, и исследования достоверности получаемых на ЭВМ решений.

III.1. НЕКОТОРЫЕ ОБЩИЕ СВЕДЕНИЯ ИЗ ЛИНЕЙНОЙ АЛГЕБРЫ

1. Постановка задачи. Задачи на собственные значения возникают при факторном анализе, при определении частот собственных колебаний консервативных и неконсервативных динамических систем, при исследовании колебаний и устойчивости объектов механической, физической и химической природы, а также как самостоятельные математические задачи.

Обобщенная задача состоит в нахождении таких чисел λ , при которых существуют отличные от нулевого решения системы линейных алгебраических уравнений

$$Av = \lambda Bv, \quad (III.1)$$

где A и B — некоторые квадратные матрицы порядка n . Числа λ называются собственными значениями, а векторы v собственными векторами обобщенной проблемы.

Конечные собственные значения задачи (III.1) есть корни уравнения

$$\det(A - \lambda B) = 0.$$

Если $\det(B) = 0$ и если $\det(A - \lambda B)$ не является нулевым многочленом, то уравнение $\det(A - \lambda B) = 0$ имеет степень, меньшую значения n . Недостающие собственные значения называются бесконечными собственными значениями и фактически являются нулевыми корнями характеристического уравнения $\det(B - \mu A) = 0$.

Если в (III.1) матрица B единичная, т. е. $B = E$, то вместо (III.1) рассматривают задачу

$$Av = \lambda v \quad (III.2)$$

и имеют в виду собственные числа и собственные векторы матрицы A (см., например, п. 10 и 13 параграфа II.1).

Собственные числа, как корни нелинейного алгебраического уравнения (II.22) или (II.23) (см. п. 10 параграфа II.1), могут быть действительными или комплексными, простыми или кратными. Одному собственному числу может соответствовать несколько линейно независимых собственных векторов.

Пример 1. Найти все собственные числа и собственные векторы матрицы

$$A = \begin{bmatrix} 5 & -2 & 1 \\ 3 & -2 & 3 \\ 3 & -6 & 7 \end{bmatrix}. \quad (III.3)$$

Решение. Составим определитель $|A - \lambda E|$ и раскроем его:

$$\begin{vmatrix} 5-\lambda & -2 & 1 \\ 3 & -2-\lambda & 3 \\ 3 & -6 & 7-\lambda \end{vmatrix} = -\lambda^3 + 10\lambda^2 - 32\lambda + 32.$$

Собственные числа являются корнями уравнения

$$\lambda^3 - 10\lambda^2 + 32\lambda - 32 = (\lambda - 2)(\lambda - 4)^2 = 0, \quad \lambda_1 = 2, \quad \lambda_2 = \lambda_3 = 4.$$

Для нахождения собственного вектора, соответствующего собственному значению $\lambda_1 = 2$, получаем систему линейных алгебраических уравнений с определителем, равным нулю:

$$\begin{aligned} 3v_1 - 2v_2 + v_3 &= 0, \\ 3v_1 - 4v_2 + 3v_3 &= 0, \\ 3v_1 - 6v_2 + 5v_3 &= 0. \end{aligned} \quad (III.4)$$

Для решения этой системы определяем ранг ее матрицы. Нетрудно убедиться, что ранг равен двум. Придавая свободному неизвестному v_1 значение 1, решаем систему первых двух уравнений

$$\begin{aligned} -2v_2 + v_3 &= -3, \\ -4v_2 - 3v_3 &= -3 \end{aligned}$$

по правилу Крамера. В результате получаем $v_2 = 3$, $v_3 = 3$. Таким образом, собственному числу $\lambda_1 = 2$ соответствует собственный вектор $v^{(1)} = \begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix}$. Для нахождения собственных векторов, соответствующих кратному собственному числу $\lambda_2 = 4$, получаем систему

$$\begin{aligned} v_1 - 2v_2 + v_3 &= 0, \\ 3v_1 - 6v_2 + 3v_3 &= 0, \\ 3v_1 - 6v_2 + 3v_3 &= 0. \end{aligned}$$

Ранг ее равен единице, поэтому решение представляем в виде

$$v_1 = 2v_2 - v_3.$$

Свободным неизвестным v_2 и v_3 нужно придать такие значения, при которых получаются два линейно независимых вектора. Для этой цели возьмем следующие значения:

$$\begin{aligned} v_2 &= 1, \quad v_3 = 0, \\ v_2 &= 0, \quad v_3 = 1 \end{aligned}$$

(см. п. 8 параграфа II.1).

Следовательно, собственному числу $\lambda_2 = \lambda_3 = 4$ соответствуют два собственных вектора:

$$v^{(2)} = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}, \quad v^{(3)} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}.$$

Задача нахождения всех собственных чисел и принадлежащих им собственных векторов называется полной проблемой собственных значений. Задача нахождения нескольких собственных чисел и соответст-

вующих им векторов или только собственных значений называется частичной проблемой собственных значений.

2. Некоторые свойства собственных значений симметричной трехдиагональной матрицы. Трехдиагональной называется матрица вида

$$A = \begin{bmatrix} a_1 & b_1 & 0 & \dots & 0 & 0 & 0 \\ c_1 & a_2 & b_2 & \dots & 0 & 0 & 0 \\ 0 & c_2 & a_3 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & c_{n-2} & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & \dots & 0 & c_{n-1} & a_n \end{bmatrix}. \quad (\text{III.5})$$

Трехдиагональные матрицы возникают при описании или решении некоторых прикладных задач. Кроме того, задачи на собственные значения для симметричных трехдиагональных матриц иногда являются частью решения задачи о нахождении собственных значений произвольных симметричных матриц. Естественно, что задача нахождения собственных значений симметричных трехдиагональных матриц проще, чем задача нахождения собственных значений произвольной симметричной матрицы.

Если для вещественной трехдиагональной матрицы выполняется неравенство $b_i c_i > 0$ для $i = 1, 2, \dots, n-1$, то такая матрица называется якобиевой (или матрицей Якоби).

Обозначив главный минор порядка r матрицы $(A - \lambda E)$ через $p_r(\lambda)$ и приняв $p_0(\lambda)$ равным единице, будем иметь

$$\begin{aligned} p_1(\lambda) &= a_1 - \lambda, \\ &\dots \dots \dots \\ p_i(\lambda) &= (a_i - \lambda) p_{i-1}(\lambda) - b_{i-1} c_{i-1} p_{i-2}(\lambda). \end{aligned} \quad (\text{III.6})$$

А для трехдиагональных симметричных матриц получим

$$p_i(\lambda) = (a_i - \lambda) p_{i-1}(\lambda) - b_{i-1}^2 p_{i-2}(\lambda), \quad i = 2, 3, \dots \quad (\text{III.7})$$

Для матрицы Якоби полиномы $p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)$ составляют последовательность Штурма [97]. Корни таких полиномов вещественны и различны.

Используя свойства последовательности Штурма, можно определить число корней характеристического полинома, больших некоторого наперед заданного числа.

Пусть величины $p_0(\mu), p_1(\mu), \dots, p_n(\mu)$ вычисляются для некоторого значения μ . Тогда число $s(\mu)$ (число совпадений знаков у соседних членов этой последовательности) равно числу тех собственных значений матрицы A , которые строго больше значения μ . Если получится так, что $p_i(\mu) = 0$, то знак нуля берут противоположным знаком значения $p_{i-1}(\mu)$. Заметим, что никакие два последовательных значения $p_i(\mu)$ не могут быть равными нулю. Пусть, например, задана

матрица

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & - & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}.$$

Для нее полиномы $p_r(\lambda)$ имеют вид

$$\begin{aligned} p_0(\lambda) &= 1, \\ p_1(\lambda) &= 2 - \lambda, \\ p_2(\lambda) &= (2 - \lambda) p_1(\lambda) - 1, \\ p_3(\lambda) &= (2 - \lambda) p_2(\lambda) - p_1(\lambda), \\ p_4(\lambda) &= (2 - \lambda) p_3(\lambda) - p_2(\lambda). \end{aligned}$$

При $\mu = 2$ эти полиномы принимают следующие значения:

$$\begin{aligned} p_0(2) &= 1, \\ p_1(2) &= -0, \\ p_2(2) &= -1, \\ p_3(2) &= +0, \\ p_4(2) &= +1. \end{aligned}$$

В полученной нами последовательности дважды изменяются знак и число совпадений знаков $s(2) = 2$. Значит, матрица A имеет два собственных значения, строго больших, чем 2.

3. Ортогональные матрицы. Действительная матрица называется ортогональной, если ее транспонированная матрица совпадает с обратной A^{-1} , т. е.

$$A^T = A^{-1}, \quad (\text{III.8})$$

или

$$AA^T = A^T A = E. \quad (\text{III.9})$$

Ортогональная матрица обладает следующими свойствами.

Строки (столбцы) ортогональной матрицы попарно ортогональны. Действительно, для ортогональной матрицы A из равенства (III.9) имеем

$$\sum_{k=1}^n a_{ik} a_{jk} = 0 \text{ при } i \neq j \text{ и } \sum_{k=1}^n a_{ki} a_{kj} = 0 \text{ при } i \neq j.$$

Сумма квадратов элементов каждой строки (столбца) ортогональной матрицы равна единице. Действительно, из равенства (III.9) получим

$$\sum_{i=1}^n a_{ij}^2 = 1, \quad j = 1, 2, \dots, n, \quad \sum_{j=1}^n a_{ij}^2 = 1, \quad i = 1, 2, \dots, n.$$

Определитель ортогональной матрицы равен ± 1 . Действительно, на основании равенства (III.9) имеем $|A| |A^T| = |E|$. Так как

$A^T = |A|$ и $|E| = 1$, то $|A|^2 = 1$ и, следовательно, $|A| = \pm 1$.

Если матрица A ортогональна, то и матрица A^{-1} тоже ортогональна.

Произведение двух ортогональных матриц является ортогональной матрицей. Действительно, если A и B ортогональны, то

$$(AB)^T AB = B^T A^T AB = B^T EB = E.$$

Если матрица A_N ортогональна, то матрица

$$A_{N+1} = \begin{bmatrix} 1 & 0 \\ 0 & A_N \end{bmatrix}$$

тоже ортогональна. Примером ортогональной матрицы может служить матрица второго порядка

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

Ортогональные матрицы так же, как и симметричные, входят в более общий класс так называемых нормальных матриц.

Действительная матрица называется нормальной, если она перестановочна со своей транспонированной, т. е. если

$$A^T A = A A^T.$$

4. Элементарные матрицы вращения. К классу ортогональных принадлежат элементарные матрицы вращения вида

$$P_{ij} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & c & & \dots & -s \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & \ddots \\ & & s & \dots & & c \\ & & & & & & \ddots \\ & & & & & & & 1 \end{bmatrix}$$

где $c^2 + s^2 = 1$.

В дальнейшем элементарные матрицы вращения будут неоднократно использоваться как вспомогательные для преобразования данной

матрицы A посредством цепочки умножений слева или справа, или с обеих сторон на эти матрицы.

Очевидно, что при умножении матрицы A слева на матрицу P_{ij} изменяются лишь i -я и j -я строки матрицы A , а именно: для матрицы $A^{(1)} = P_{ij}A$ будем иметь

$$a_{i\beta}^{(1)} = ca_{i\beta} - sa_{j\beta},$$

$$a_{j\beta}^{(1)} = sa_{i\beta} + ca_{j\beta}, \quad \beta = 1, 2, \dots, n. \quad (\text{III.10})$$

При умножении матрицы A справа на матрицу P_{ij} изменяются только i -й и j -й столбцы по формулам

$$a_{\alpha i}^{(1)} = ca_{\alpha i} + sa_{\alpha j}$$

$$a_{\alpha j}^{(1)} = -sa_{\alpha i} + ca_{\alpha j}, \quad \alpha = 1, 2, 3, \dots, n. \quad (\text{III.11})$$

Ясно, что если хотя бы один из элементов $a_{i\beta}$ и $a_{j\beta}$ отличен от нуля, то можно подобрать c и s так, чтобы для матрицы $A^{(1)} = P_{ij}A$ элемент $a_{j\beta}^{(1)}$ оказался равным нулю. Для этого нужно взять

$$s = \frac{a_{j\beta}}{\sqrt{a_{i\beta}^2 + a_{j\beta}^2}}, \quad c = \frac{a_{i\beta}}{\sqrt{a_{i\beta}^2 + a_{j\beta}^2}}. \quad (\text{III.12})$$

При таком выборе s и c получаем

$$a_{ij}^{(1)} = \sqrt{a_{i\beta}^2 + a_{j\beta}^2} > 0, \quad a_{j\beta}^{(1)} = 0. \quad (\text{III.13})$$

Получение в матрице $A^{(1)} = P_{ij}A$ на месте элемента $a_{j\beta}^{(1)}$ нуля (аннулирование элемента $a_{j\beta}^{(1)}$) является полезной особенностью матриц вращения, которая часто будет использоваться в дальнейшем. Отметим следующее.

Любая действительная невырожденная матрица посредством цепочки умножений слева на элементарные матрицы вращения может быть преобразована в правую треугольную матрицу, все диагональные элементы которой, возможно, кроме последнего, положительны, т. е.

$$A^{(n-1)} = P_{n-1,n} \dots P_{12}A = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn}^{(n-1)} \end{bmatrix}.$$

Всякая невырожденная действительная матрица есть произведение ортогональной матрицы на правую треугольную. Действительно, $A = PA^{(n-1)}$, где $P = (P_{n-1,n} \dots P_{12})^{-1}$ ортогональная матрица.

Всякая ортогональная матрица с определителем, равным единице, есть произведение элементарных матриц вращения, т. е. если A ортогональная матрица, то

$$A = P_{12}^{-1} \dots P_{n-1,n}^{-1}E.$$

Для действительной симметричной матрицы A существует такая ортогональная матрица P , что

$$P^{-1}AP = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix},$$

где λ_i — собственные значения матрицы A , причем столбцы матрицы P являются собственными векторами матрицы A .

Использование в преобразованиях подобия (см. п. 11 параграфа II.1) ортогональных матриц вращения позволит, с одной стороны, привести исходную матрицу A к такой матрице, собственные значения которой находятся сравнительно просто (диагональной, треугольной, трехдиагональной и т. д.), с другой — вместо операции обращения матрицы использовать транспонирование, что намного упрощает вычисления.

5. Матрицы отражения. Другим примером ортогональной матрицы является матрица отражения Q , которая воздействует на некоторый вектор z , отражает его на вектор l единичной длины с коэффициентом α (см. п. 8 параграфа I.1), т. е.

$$Qz = \alpha l, \quad (\text{III.14})$$

где $\alpha = \pm \sqrt{(z, z)}$.

Матрица отражения Q имеет вид

$$Q = E - 2ww^T, \quad \|w\| = 1. \quad (\text{III.15})$$

Здесь

$$w = \frac{1}{\rho} (z - \alpha l), \quad \rho^2 = 2[\alpha^2 - \alpha(z, l)],$$

а E — единичная матрица.

Матрица Q симметрична и ортогональна. Симметричность ее очевидна, а ортогональность легко показать:

$$\begin{aligned} QQ^T &= Q^TQ = (E - 2ww^T)(E - 2ww^T) = \\ &= E - 4ww^T + 4w(w^Tw)w^T = E. \end{aligned}$$

Приведем пример построения матрицы отражения по вектору $z = [1, 2, 3, 4]^T$. После нормировки этот вектор приобретает вид $\left[\frac{1}{30}, \frac{2}{30}, \frac{3}{30}, \frac{4}{30}\right]^T$. И матрица Q , построенная по формуле (III.15), записы-

вается следующим образом:

$$Q = E - 2 \begin{bmatrix} \frac{1}{30} & \frac{2}{30} & \frac{3}{30} & \frac{4}{30} \\ \frac{2}{30} & \frac{4}{30} & \frac{6}{30} & \frac{8}{30} \\ \frac{3}{30} & \frac{6}{30} & \frac{9}{30} & \frac{12}{30} \\ \frac{4}{30} & \frac{8}{30} & \frac{12}{30} & \frac{16}{30} \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{28}{30} & \frac{4}{30} & \frac{6}{30} & \frac{8}{30} \\ -\frac{4}{30} & \frac{22}{30} & -\frac{12}{30} & -\frac{16}{30} \\ -\frac{6}{30} & -\frac{12}{30} & \frac{12}{30} & -\frac{24}{30} \\ -\frac{8}{30} & -\frac{16}{30} & -\frac{24}{30} & \frac{2}{30} \end{bmatrix}.$$

Следует отметить, что при преобразовании отражения нет необходимости строить матрицу отражения в явном виде; она определяется вектором w , а он отличается от z лишь одним элементом.

6. Каноническая форма Жордана. Если матрица A имеет n различных собственных значений, то преобразованием подобия ее можно привести к диагональной форме. Однако при наличии кратных корней такое преобразование возможно не всегда. Тем не менее можно поставить задачу о приведении матрицы путем преобразования подобия к наиболее простому виду. Эта задача равнозначна отысканию такого базиса, в котором линейное преобразование, связанное с данной матрицей, имело бы матрицу простейшей формы. Такой простейшей формой оказывается так называемая каноническая форма Жордана.

Отметим, что всякая матрица посредством преобразования подобия (см. п. 11 параграфа II.1) может быть приведена к канонической форме Жордана. Опишем ее.

Каноническим ящиком называется матрица следующего вида:

$$\begin{bmatrix} \lambda_i & 0 & 0 & \dots & 0 & 0 \\ 1 & \lambda_i & 0 & \dots & 0 & 0 \\ 0 & 1 & \lambda_i & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & \lambda_i \end{bmatrix}.$$

Канонический ящик не может быть упрощен за счет преобразования подобия. Очевидно, что он имеет единственное кратное собственное число λ_i . Легко проверить, что канонический ящик имеет только один с точностью до умножения на ненулевое число собственный вектор. Характеристический полином канонического ящика имеет вид

$$(\lambda_i - \lambda)^{m_i},$$

где m_i — порядок ящика.

Матрицей в канонической форме Жордана называется квазидиagonalная матрица, составленная из канонических ящиков, при этом допускается, чтобы одно и то же число λ_i входило в несколько канонических ящиков

$$\begin{bmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 1 & \lambda_1 & 0 & \dots & 0 \\ 0 & 1 & \lambda_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_1 \\ & & & & \vdots \\ & & & & \lambda_s & 0 & 0 & \dots & 0 \\ & & & & 1 & \lambda_s & 0 & \dots & 0 \\ & & & & 0 & 1 & \lambda_s & \dots & 0 \\ & & & & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & 0 & 0 & 0 & \dots & \lambda_s \end{bmatrix}.$$

Все числа λ_i , входящие в различные канонические ящики, являются собственными числами канонической матрицы, причем кратность собственного числа равна сумме порядков ящиков, в которые оно входит в качестве диагонального элемента.

Пусть, например, матрица девятого порядка имеет трехкратное собственное значение λ_1 , которому соответствует один собственный вектор; два простых собственных значения λ_4 и λ_5 и четырехкратное собственное значение λ_6 , которому соответствуют три линейно независимых вектора.

Каноническая форма Жордана такой матрицы имеет вид

$$\begin{bmatrix} \lambda_1 & 0 & 0 & & & & & & \\ 1 & \lambda_1 & 0 & & & & & & \\ 0 & 1 & \lambda_1 & & & & & & \\ & & & \lambda_4 & & & & & \\ & & & & \lambda_5 & & & & \\ & & & & & \lambda_6 & & & \\ & & & & & & \lambda_6 & & \\ & & & & & & & \lambda_6 & 0 \\ & & & & & & & 1 & \lambda_6 \end{bmatrix}.$$

Каноническая форма Жордана для матрицы, обладающей полным набором линейно независимых собственных векторов, есть диагональная матрица.

7. Элементарные делители. Пусть задана матрица A шестого порядка, которая имеет пятикратное собственное значение λ_1 , однократное собственное значение λ_2 и ее жорданова каноническая форма J представима в виде

$$J = \begin{bmatrix} J_2(\lambda_1) & & & \\ & J_2(\lambda_1) & & \\ & & J_1(\lambda_1) & \\ & & & J_1(\lambda_2) \end{bmatrix}, \quad (\text{III.16})$$

где $J_r(\lambda_i)$ — канонические ящики r -го порядка, $i = 1, 2$.

Рассмотрим матрицу

$$J - \lambda E = \begin{bmatrix} J_2(\lambda_1 - \lambda) & & & \\ & J_2(\lambda_1 - \lambda) & & \\ & & J_1(\lambda_1 - \lambda) & \\ & & & J_1(\lambda_2 - \lambda) \end{bmatrix}.$$

Ясно, что $J - \lambda E$ есть прямая сумма матриц $J_r(\lambda_i - \lambda)$. Определители канонических ящиков канонической формы $J - \lambda E$ называются элементарными делителями матрицы A . Так, элементарные делители любой матрицы, подобной J из (III.16), будут

$$(\lambda_1 - \lambda)^2, (\lambda_1 - \lambda)^2, (\lambda_1 - \lambda), (\lambda_2 - \lambda).$$

Очевидно, что произведение элементарных делителей матрицы равно ее характеристическому полиному. Если каноническая форма Жордана диагональна, то элементарными делителями будут

$$(\lambda_1 - \lambda), (\lambda_2 - \lambda), \dots, (\lambda_n - \lambda),$$

так что все они линейны. У матрицы с различными собственными значениями элементарные делители всегда линейны. Если у матрицы есть одно или более кратное собственное значение, то она может иметь или не иметь нелинейные элементарные делители.

Если у матрицы есть один или более нелинейный элементарный делитель, то, по крайней мере, один из канонических ящиков в ее канонической форме Жордана имеет порядок, больший единицы, и, следовательно, у нее будет меньше, чем n , линейно независимых собственных векторов.

III.2. ОБУСЛОВЛЕННОСТЬ МАТРИЦ В ЗАДАЧАХ НА СОБСТВЕННЫЕ ЗНАЧЕНИЯ

1. Постановка вычислительных задач на собственные значения. При исследовании прикладных проблем, описываемых задачами на собственные значения, могут быть поставлены следующие теоретические задачи.

Для плотной или ленточной действительной симметричной матрицы A найти некоторые или все ее собственные значения, а иногда и соответствующие им собственные векторы. Аналогичная задача ставится в некоторых случаях для комплексной эрмитовой матрицы¹¹. В этом случае каждое собственное значение λ является действительным, а собственные векторы v могут быть комплексными.

В ряде случаев требуется найти все или некоторые собственные значения, а иногда принадлежащие им собственные векторы обобщенной задачи на собственные значения с действительной и симметричной матрицей A и с действительной симметричной положительно определенной матрицей B . Матрицы A и B могут быть плотными или ленточными.

В некоторых случаях для произвольной действительной или комплексной матрицы A необходимо найти все или некоторые собственные значения и иногда принадлежащие им собственные или корневые векторы¹². Отметим, что даже для действительных матриц A все λ могут быть комплексными.

Существуют и иные постановки задач на собственные значения. В настоящей главе рассмотрим лишь некоторые из них.

При описании прикладных задач на собственные значения крайне редко возникают индивидуально задаваемые системы

$$\tilde{A}\tilde{v} = \tilde{\lambda}\tilde{v}. \quad (\text{III.17})$$

Наиболее типичным является задание системы

$$Av = \lambda v \quad (\text{III.18})$$

и погрешности в исходных данных

$$\|\tilde{A} - A\| = \|\Delta A\| \leq \varepsilon. \quad (\text{III.19})$$

Не всегда близость элементов A и $\tilde{A}$ обеспечивает близость собственных значений матрицы. Так, собственными значениями матрицы

$$\begin{bmatrix} 1 & 10\,000 \\ 0 & 1 \end{bmatrix}$$

¹¹ Комплексной эрмитовой матрицей A называется матрица, для которой $A^H = A$, где $A^H = \bar{A}^T$, $\bar{A}$ — матрица с элементами, комплексно-сопряженными с элементами матрицы A .

¹² Если в задаче (III.2) λ имеет кратность m , большую единицы, а v — вектор, отвечающий λ , то не обязательно существуют еще какие-либо собственные векторы, соответствующие λ и линейно независимые от v . Если вектор v является единственным с точностью до пропорциональности собственным вектором, принадлежащим λ , то существуют так называемые корневые векторы $y_1, y_2, \dots, y_{m-1}$ (не однозначно определенные) такие, что

$$\begin{aligned} Av &= \lambda v, \\ Ay_1 &= \lambda y_1 + v, \\ &\vdots \\ Ay_m &= \lambda y_m + y_{m-1}. \end{aligned}$$

являются $\lambda_1 = 1, \lambda_2 = 1$, а возмущенной матрицы

$$\begin{bmatrix} 1 & 10\,000 \\ 0,0001 & 1 \end{bmatrix}$$

$\lambda_1 = 2, \lambda_2 = 0$ [97].

При вычислении собственных векторов матриц также возникает задача оценки достоверности получаемых решений. Рассмотрим простой пример. Пусть заданная матрица имеет лишь простые собственные значения. Однако при некотором определенном изменении ее элементов в пределах точности их задания можно получить матрицу, имеющую кратные собственные значения. В этом случае каноническая форма матрицы при изменении ее элементов в пределах точности их задания может перейти от чисто диагональной формы к недиагональной канонической форме Жордана [103]. При этом изменяется даже количество линейно независимых собственных векторов.

На основании изложенных выше рассуждений видно, что прикладные задачи при сведении их к задачам на собственные значения порождают целый ряд уравнений (III.18), (III.19), обладающих достаточно широким классом формально допустимых решений. Решение всей проблемы будет состоять в определении одного из допустимых решений, получении этого решения и оценке наследственной погрешности в математическом решении задачи.

2. Обусловленность матриц при нахождении собственных значений. Для измерения чувствительности собственных значений λ к изменениям исходных данных ΔA вводят понятие обусловленности матрицы аналогично тому, как это сделано в п. 4 параграфа 1.2. Однако в проблеме собственных значений понятие обусловленности матрицы не совпадает с таковым при решении систем линейных алгебраических уравнений. Плохо обусловленная относительно решения системы линейных алгебраических уравнений матрица может быть хорошо обусловлена относительно собственных значений, и наоборот.

Если A — матрица с линейными элементарными делителями и ее собственные значения обозначены через $\lambda_i, i = 1, 2, \dots, n$, а λ — собственное значение матрицы $A + \Delta A$, то существует оценка [97]:

$$\min_i |\lambda_i - \lambda| \leq \|H^{-1}\|_2 \|H\|_2 \|\Delta A\|_2,$$

где H — матрица, составленная из собственных векторов A . Из этой формулы очевидно, что наследственная погрешность определяется как погрешностью в исходных данных, так и величиной

$$K(H) = \|H^{-1}\|_2 \|H\|_2, \quad (\text{III.20})$$

которую называют спектральным числом обусловленности матрицы A . Это число характеризует обусловленность матрицы при нахождении всех собственных значений в целом и в случае ее плохой обусловленности не всегда свидетельствует о плохой обусловленности отдельного собственного значения.

Для измерения чувствительности отдельного собственного значения к изменениям исходных данных было введено понятие κ чисел, которые определяются по формуле

$$s_i = y_i^T x_i,$$

где x_i — нормированный собственный вектор, соответствующий собственному значению λ_i в матрице A , y_i — нормированный собственный вектор, соответствующий λ_i в матрице A^T .

Числа $|s_i^{-1}|$ называют числами обусловленности собственных значений λ_i матрицы A и обозначают

$$\text{cond } \lambda_i = \frac{1}{|y_i^T x_i|}, \quad i = 1, 2, \dots, n. \quad (\text{III.21})$$

Если собственные векторы вещественны, то

$$\text{cond } \lambda_i = \frac{1}{|\cos(y_i, x_i)|}, \quad i = 1, 2, \dots, n.$$

Отметим, что для клетки Жордана $A = \begin{bmatrix} a & 0 \\ 1 & a \end{bmatrix}$ собственный вектор $x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, а для $A^T = \begin{bmatrix} a & 1 \\ 0 & a \end{bmatrix}$ собственный вектор $y = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ и,

следовательно, $\cos(x, y) = 0$; $\text{cond } \lambda_i = |s_i^{-1}| = \infty$. Отсюда следует, что кратные собственные значения, которым соответствует один собственный вектор, имеют плохо обусловленную матрицу при нахождении собственных значений. Однако могут быть плохо обусловленные матрицы, когда все собственные значения различны.

Если матрица A обладает простыми собственными значениями, то число обусловленности для любого из них может быть определено по формуле

$$\text{cond } \lambda = \frac{\|y\| \|x\|}{|y^T x|}. \quad (\text{III.22})$$

Здесь $\|x\|$ — евклидова длина вектора x . В этом случае, если матрица $(A + \Delta A)$ близка к A , справедлива оценка [103]

$$|\tilde{\lambda} - \lambda| = |\Delta \lambda| \leq \text{cond } \lambda \|\Delta A\|. \quad (\text{III.23})$$

3. Обусловленность матриц при нахождении собственных векторов. Существует ряд задач, в которых требуется определить не только собственные значения, но и собственные векторы. Критерий хорошей или плохой обусловленности матрицы при нахождении собственных значений не будет совпадать с критерием хорошей или плохой обусловленности матрицы при вычислении собственных векторов. Так, любая матрица (в том числе и симметричная), имеющая близкие собственные

значения, может быть хорошо обусловлена относительно вычисления собственных значений и плохо обусловлена с точки зрения нахождения собственных векторов, соответствующих близким собственным значениям.

Если собственные значения матрицы A различны и матрица A близка к $A + \Delta A$, то справедлива следующая оценка:

$$\frac{\|\Delta v_i\|}{\|v_i\|} = \frac{\|\tilde{v}_i - v_i\|}{\|v_i\|} \leq \sum_{\substack{j=1 \\ j \neq i}}^n \frac{\text{cond } \lambda_j}{|\lambda_i - \lambda_j|} \|\Delta A\|. \quad (\text{III.24})$$

Возмущение ΔA может нарушить кратность собственных значений, образуя «гроздь» собственных значений, близких друг к другу.

4. Обусловленность при вычислении собственных значений и собственных векторов нормальных матриц. Очень важным классом матриц, встречающихся в приложениях, являются нормальные матрицы¹³ и, в частности, действительные симметричные матрицы. Поэтому исследование устойчивости этого класса матрицы при решении задач на собственные значения приобретает важное практическое значение.

Если матрица A вещественна и симметрична, то матрица нормированных собственных векторов H в (III.20) ортогональна и ее спектральная норма равна единице. Отсюда следует, что вычисление собственных значений для симметричных матриц всегда устойчиво и близость математического и физического решений определяется лишь погрешностью в исходных данных

$$|\Delta \lambda| \leq \|\Delta A\|.$$

Если в симметричной матрице все собственные значения различны, то из (III.24) следует оценка относительной погрешности в собственных векторах такой матрицы:

$$\frac{\|\Delta v_i\|}{\|v_i\|} \leq \|\Delta A\| \sum_{\substack{j=1 \\ j \neq i}}^n \frac{1}{|\lambda_i - \lambda_j|},$$

которая зависит как от нормы погрешности, так и от близости собственных значений λ_i и λ_j .

5. Погрешность машинной реализации алгоритмов. Основными вопросами машинной реализации численных методов являются получение на ЭВМ решения, а также обеспечение оценки близости машинного и математического решений задачи. Машинная реализация методов нахождения собственных значений и собственных векторов задачи (III.18), (III.19) вносит погрешность, определяемую свойствами матрицы A , методами решения проблемы собственных значений и особенностями вычислений на ЭВМ.

¹³ Матрицы, для которых $AA^H = A^H A$, называются нормальными. В класс нормальных матриц входят эрмитовы матрицы ($A^H = A$), косоэрмитовы ($A^H = -A$), унитарные ($A^H = A^{-1}$), действительные симметричные ($A^T = A$).

Вычисленные на ЭВМ собственные значения являются точными собственными значениями матрицы $A + dA$, причем матрица dA не единственная и определяется выбранным алгоритмом, порядком A и длиной мантиссы машинного слова.

Теоретически при вычислении на ЭВМ всех собственных значений можно использовать следующую апостериорную оценку близости машинного и математического решений задачи [97].

$$\min_i |\mu_i - \lambda_i| \leq K(H) \frac{\|\theta\|_2}{\|v_i\|_2},$$

где μ_i — машинное приближение к собственному значению λ_i матрицы A , v_i — машинное приближение к собственному вектору, соответствующему λ_i , $K(H)$ — спектральное число обусловленности, определяемое по формуле (III.20), θ — невязка задачи на собственные значения, $\theta = Av_i - \mu_i v_i$.

Рассматривая погрешность машинной реализации в задачах на собственные значения как возмущение матрицы dA , можно выписать оценки возмущений собственных чисел и векторов при вычислениях, аналогичные оценкам, рассмотренным в п. 2—4 настоящего параграфа. Так как значение dA определяется порядком матрицы, методом решения задачи на собственные значения и длиной машинного слова, то увеличивая длину машинного слова, можно улучшить точность вычисленных собственных значений и получить достаточную близость машинного и математического решений задачи.

На основании изложенного выше видно, что при описании физических моделей существует целый класс задач на собственные значения (III.18), (III.19). Вопросы влияния возмущений в исходных данных матрицы на решение проблемы собственных значений достаточно подробно рассмотрены в гл. 2 книги [97]. Близость математического и физического решений задачи определяется, с одной стороны, свойствами матрицы задачи, с другой — погрешностью в задании исходных данных. Учет обусловленности задач при отыскании собственных значений и собственных векторов позволяет оценить наследственную погрешность в математическом решении задачи.

Погрешность машинной реализации в задачах на собственные значения можно трактовать как возмущение исходных данных. Приблизить машинное решение к математическому можно, увеличивая длину машинного слова. Учет погрешности вычислений на ЭВМ является одним из аспектов машинной реализуемости алгоритмов. При этом представляют интерес простота вычислительных схем, оценка погрешности реализации каждого метода, требуемого числа арифметических операций и памяти машины и в конечном итоге время решения задачи на машине.

III.3. НЕКОТОРЫЕ МЕТОДЫ РЕШЕНИЯ ПОЛНОЙ ПРОБЛЕМЫ СОБСТВЕННЫХ ЗНАЧЕНИЙ

1. Предварительные сведения. В основе большинства методов вычисления всех собственных значений и принадлежащих им собственных векторов лежит преобразование подобия матриц.

Преобразованием подобия называют (см. п. 11 параграфа II.1) преобразование вида

$$C = T^{-1}AT, \quad |T| \neq 0. \quad (\text{III.25})$$

В этом случае матрицу C называют подобной матрице A . Преобразованием подобия исходную матрицу A приводят к матрице C , у которой собственные значения вычисляются проще, чем у исходной матрицы A . В качестве матрицы T удобно использовать ортогональные матрицы (см. п. 3 параграфа I.3), так как для них обратная матрица совпадает с транспонированной и нет необходимости ее вычислять. И что еще важнее — преобразования с ними численно устойчивы. Среди ортогональных матриц весьма интересны матрицы вращения, позволяющие, с одной стороны, путем подбора параметров c и s уничтожить (аннулировать) соответствующий элемент при произведении матрицы вращения на преобразуемую матрицу, с другой — сводить умножение на матрицу вращения слева или справа к пересчету двух строк или двух столбцов преобразуемой матрицы по простым формулам. Различные методы, основанные на применении матриц вращения, сходят в разном способе аннулирования элементов матрицы, для которой разыскиваются собственные значения и собственные векторы.

2. Метод Якоби. Для вычисления всех собственных значений и принадлежащих им собственных векторов симметричной матрицы целесообразно использовать метод Якоби.

Перед описанием метода Якоби напомним, что всякая симметричная матрица A n -го порядка может быть представлена в виде

$$A = Q^T \Lambda Q, \quad (\text{III.26})$$

где Q — ортогональная матрица, Λ — диагональная с элементами $\lambda_1, \lambda_2, \dots, \lambda_n$, которые являются собственными значениями матрицы A . Из (III.26) можно получить

$$AQ^T = Q^T \Lambda, \quad QAQ^T = \Lambda.$$

Из приведенных равенств видно, что i -я строка матрицы Q является собственным вектором, соответствующим собственному числу λ_i . Таким образом, преобразование подобия переводит матрицу A в диагональную Λ .

В методе Якоби матрица A , для которой разыскиваются собственные значения и принадлежащие им собственные векторы, преобразуется в диагональную матрицу Λ с помощью цепочки преобразований вращения

$$\Lambda = \dots P_k \dots P_2 P_1 A_0 P_1^T P_2^T \dots P_k^T \dots, \quad A_0 = A,$$

причем на k -м шаге процесс преобразования осуществляется по формуле

$$A_k = P_k A_{k-1} P_k^T, \quad (\text{III.27})$$

[illegible]

Измененные элементы вычисляются по формулам

$$\begin{aligned} a_{ip}^{(k)} &= a_{ip}^{(k-1)}c + a_{iq}^{(k-1)}s = a_{pi}^{(k)}, \\ a_{iq}^{(k)} &= -a_{ip}^{(k-1)}s + a_{iq}^{(k-1)}c = a_{qi}^{(k)}, \quad i \neq p, q, \end{aligned} \quad (\text{III.29})$$

$$\begin{aligned} a_{pp}^{(k)} &= a_{pp}^{(k-1)} s^2 + 2a_{pq}^{(k-1)} sc + a_{qq}^{(k-1)} c^2, \\ a_{qq}^{(k)} &= a_{pp}^{(k-1)} c^2 - 2a_{pq}^{(k-1)} sc + a_{qq}^{(k-1)} s^2, \end{aligned} \quad (\text{III.30})$$

$$a_{pq}^{(k)} = (a_{qq}^{(k-1)} - a_{pp}^{(k-1)})cs + a_{pq}^{(k-1)}(s^2 - c^2) = a_{qp}^{(k)} = 0.$$

Числа c и s в методе Якоби получают из условий

$$c^2 + s^2 = 1, \quad a_{pq}^{(k)} = a_{qp}^{(k)} = 0.$$

$$c = \sqrt{1 - s^2}, \quad s = \frac{\lambda}{\sqrt{2(1 - \sqrt{1 - x^2})}}, \quad x = \begin{cases} -1, & \text{если } y = 0, \\ \frac{\operatorname{sign}(y) a_{pq}}{-\sqrt{a_{pq}^2 + y^2}}, & \text{если } y \neq 0, \end{cases} \quad (\text{III.31})$$

где $y = \frac{1}{2}(a_{pp} - a_{qq})$.

На практике процесс получения диагональных элементов заканчивается тогда, когда все внедиагональные элементы матрицы A_k близки к нулю. В этот момент матрица $Q = P_1 P_2 \dots P_k$ является матрицей приближенных собственных векторов.

В методе Якоби существует вопрос о порядке аннулирования внедиагональных элементов. Удобно аннулирование элементов подряд, т. е. элементов с индексами $(1, 2), (1, 3), \dots (1, n), (2, 3), \dots, (2, n)$ и т. д. Эта схема иногда называется частным циклическим методом Якоби, в отличие от обобщенного метода, в котором все внедиагональные элементы исключаются по одному разу в некоторой последовательности из $\frac{1}{2}(n-1)n$ вращений, но не обязательно в приведенном выше порядке.

Недостатком такого метода является то, что по ходу процесса приходится аннулировать малые внедиагональные элементы, хотя в это время есть и большие. Это отрицательно сказывается на точности всего процесса.

Для достижения наибольшей точности следует на каждом шаге аннулировать максимальный внедиагональный элемент (классический метод Якоби).

Пример 2. Найти собственные значения матрицы

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

методом Якоби с аннулированием максимального по модулю внедиагонального элемента.

Решение. Наибольший внедиагональный элемент $a_{12} = 2$. Для аннулирования вычисляем элементы s и c матрицы вращения P_1 по формулам (III.31):

$$y = \frac{1}{2} (a_{11} - a_{22}) = -1, \quad x = -\operatorname{sign}(y) \frac{a_{12}}{\sqrt{a_{12}^2 + y^2}} = \frac{2}{\sqrt{5}},$$

$$s = 0,525\ 7, \quad c = 0,850\ 6.$$

Элементы матрицы $A_1 = P_1 A_0 P_1^T$ вычисляются по формулам

$$a_{13}^{(1)} = a_{31}^{(1)} = ca_{31} + sa_{32} = -0,2008,$$

$$a_{32}^{(1)} = a_{23}^{(1)} = -sa_{31} + ca_{32} = -2,227\ 0,$$

$$\begin{aligned} a_{11}^{(1)} &= s^2 a_{11} + 2sca_{12} + c^2 a_{22} = 0,236 \text{ 1}, \\ a_{22}^{(1)} &= c^2 a_{11} - 2sca_{12} + s^2 a_{22} = 4,236 \text{ 1}, \\ a_{12}^{(1)} &= a_{21}^{(1)} = -10^{-5}, \quad a_{33}^{(1)} = a_{33} = 1. \end{aligned}$$

Таким образом,

$$A_1 = \begin{bmatrix} -0,236 \text{ 1} & -10^{-5} & -0,200 \text{ 8} \\ -10^{-5} & 4,236 \text{ 1} & 2,227 \text{ 0} \\ -0,200 \text{ 8} & 2,227 \text{ 0} & 1 \end{bmatrix}.$$

Для получения матрицы собственных векторов p -ю и q -ю строки единичной матрицы преобразуют по формулам

$$\begin{aligned} t_{pk} &= ct_{pk} - st_{pk}, \\ t_{qk} &= st_{pk} + ct_{pk}, \quad k = 1, 2, \dots, n. \end{aligned}$$

и получают матрицу

$$P_1 = \begin{bmatrix} 0,850 \text{ 7} & -0,525 \text{ 7} & 0 \\ 0,525 \text{ 7} & 0,850 \text{ 6} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

На этом заканчивается первый шаг процесса.

Далее аннулируем максимальный внедиагональный элемент матрицы A_1 , им будет, как нетрудно убедиться, элемент $a_{23}^{(1)} = 2,2270$. Элементы s и c матрицы вращения P_2 в этом случае имеют значения $s = -0,454 \text{ 0}$, $c = 0,891 \text{ 0}$, а элементы матрицы $A_2 = P_2 P_1 A_0 P_1^T P_2^T$ принимают значения

$$\begin{aligned} a_{11}^{(2)} &= -0,236 \text{ 1}, \quad a_{22}^{(2)} = 5,370 \text{ 8}, \quad a_{33}^{(2)} = -0,134 \text{ 7}, \\ a_{12}^{(2)} &= -0,091 \text{ 2}, \quad a_{13}^{(2)} = -0,178 \text{ 9}, \quad a_{23}^{(2)} = -10^{-6}, \end{aligned}$$

т. е.

$$A_2 = \begin{bmatrix} -0,236 \text{ 1} & -0,091 \text{ 2} & -0,178 \text{ 9} \\ -0,091 \text{ 2} & 5,370 \text{ 8} & -10^{-6} \\ -0,178 \text{ 9} & -10^{-6} & -0,134 \text{ 73} \end{bmatrix}.$$

Матрица $P_2 P_1$ приобретает вид

$$P_2 P_1 = \begin{bmatrix} 0,850 \text{ 6} & -0,525 \text{ 7} & 0 \\ 0,468 \text{ 4} & 0,757 \text{ 9} & 0,454 \text{ 0} \\ 0,238 \text{ 7} & 0,386 \text{ 2} & 0,891 \text{ 0} \end{bmatrix}.$$

После шестого шага получаем

$$\begin{aligned} A_6 &= \begin{bmatrix} -0,372 \text{ 3} & 10^{-5} & 0 \\ 10^{-5} & 5,372 \text{ 3} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ \prod_{i=1}^6 P_i &= \begin{bmatrix} 0,541 \text{ 8} & -0,642 \text{ 6} & 0,541 \text{ 8} \\ 0,454 \text{ 4} & 0,766 \text{ 2} & 0,454 \text{ 4} \\ -0,707 \text{ 1} & 0 & 0,707 \text{ 1} \end{bmatrix}. \end{aligned}$$

Таким образом, матрица A_6 с точностью до 10^{-5} является диагональной. Ее диагональные элементы являются приближениями к собственным числам исходной матрицы $A_0 = A$, а строки матрицы $Q = \prod_{i=1}^6 P_i$ — соответствующими собственными векторами.

Показано (см., например, [97]), что классический метод Якоби всегда сходится и имеет квадратичную скорость сходимости.

Представим матрицу A_k в виде суммы двух матриц: диагональной D_k и симметричной матрицы внедиагональных элементов C_k , т. е.

$$A_k = D_k + C_k.$$

Для классического метода Якоби, в котором на каждом шаге аннулируется максимальный внедиагональный элемент, справедлива следующая оценка. Если для всех i, j

$$|\lambda_i - \lambda_j| > 2\delta, \quad i \neq j \quad (\text{III.32})$$

и через r шагов имеют

$$\|C_r\|_E < \frac{1}{r} \delta, \quad (\text{III.33})$$

то через $N = \frac{n(n-1)}{2}$ преобразований получают

$$\|C_{r+N}\|_E \leq \frac{(1/2 n - 1)^{1/2}}{\delta} \|C_r\|_E^2.$$

Для общего циклического метода Уилкинсон показал, что если выполняются неравенства (III.32), (III.33), то справедлива оценка

$$\|C_{r+N}\|_E \leq \frac{1}{2\delta} n(n-1)^{1/2} \|C_r\|_E^2, \quad (\text{III.34})$$

а для частного циклического метода —

$$\|C_{r+N}\|_E \leq \frac{1}{\delta \sqrt{2}} \|C_r\|_E^2. \quad (\text{III.35})$$

Отметим, однако, что оценки (III.34), (III.35) показывают лишь, что если эти процессы сходятся, то асимптотически они сходятся квадратично. Сходимость же частного циклического метода доказана, если углы вращения соответствующим образом ограничены. Сходимость общего циклического метода не доказана.

Следует отметить, что при реализации классического метода Якоби на ЭВМ требуется большое количество машинных операций для отыскания максимального внедиагонального элемента, поэтому при реализации метода Якоби на ЭВМ целесообразно применять аннулирование «по преградам» (см., например, [103]). В этом случае задают монотонно убывающую к нулю последовательность чисел (преград).

$$\alpha_k = \frac{\sigma}{n^{2k+1}}, \quad k = 0, 1, \dots, \quad \sigma = \sqrt{\sum_{i \neq j} a_{ij}^2}. \quad (\text{III.36})$$

Далее находят внедиагональный элемент a_{ji} , по модулю больший, чем α_0 . Если такого элемента нет, то преграда α_0 заменяется на α_1 . Если есть элемент, превышающий преграду, то он аннулируется. После аннулирования всех внедиагональных элементов матрицы, больших, чем α_k , строится следующая преграда по формуле (III.36). Процесс продолжается до тех пор, пока все внедиагональные элементы не станут меньше значения ε (значение ε задается до начала итерационного процесса). Этот процесс позволяет решать полную проблему собственных значений значительно быстрее, чем процесс с выбором максимального элемента. Однако при практическом его использовании встречается ряд трудностей, связанных с относительным выбором барьера. Если его выбрать очень большим, то будет затрачено много времени на просмотр малых элементов, а если малым, то будет затрачено много времени на исключение малых элементов, которые по существу не влияют на скорость сходимости.

Особого внимания заслуживает следующий способ выбора элемента, подлежащего исключению [15]. Если в матрице A_k исключается элемент, стоящий в позиции i_k, j_k , то суммы квадратов внедиагональных элементов в каждой строке матрицы A_{k+1} будут такие же, как матрицы A , кроме строк с номерами i_k, j_k . Поэтому если в начале процесса вычислить суммы квадратов внедиагональных элементов строк матрицы A , то в дальнейшем в полученной последовательности $\{\beta_p\}$ из n чисел при каждом элементарном шаге будут изменяться лишь два числа. Этот факт позволяет находить оптимальный для исключения элемент путем просмотра всего лишь $2n - 1$ чисел. Определяется он следующим образом. Сначала в последовательности $\{\beta_p\}$ находят максимальный элемент β_{i_k} , а затем в i_k -й строке отыскивают максимальный по модулю элемент a_{i_k, j_k} . Очевидно, что он будет близок к наибольшему по модулю и во всяком случае не меньше, чем среднее квадратичное всех внедиагональных элементов. Для того чтобы подготовить последовательность $\{\beta_p\}$ к следующему шагу, необходимо пересчитать числа β_{i_k} и β_{j_k} .

Вся теория метода вращений с выбором максимального элемента полностью переносится на процесс с выбором оптимального элемента.

Что касается оценки точности методов Якоби при их численной реализации, то следует отметить, что в работе [97] получены мажорантная оценка близости машинного решения к математическому

$$\left[\frac{\sum (\mu_i - \lambda_i)^2}{\sum \lambda_i^2} \right]^{1/2} \leq 108 \cdot 2^{-t} r n^{3/2} (1 + 9 \cdot 2^{-t})^{6(4n-7)}$$

и вероятностная оценка

$$\left[\frac{\sum (\mu_i - \lambda_i)^2}{\sum \lambda_i^2} \right]^{1/2} \leq 11 \cdot 2^{-t} n^{3/4}.$$

В этих оценках μ_i — диагональные элементы конечной матрицы процесса, λ_i — точные собственные значения исходной матрицы A ($i = 1, 2, \dots, n$), t — число двоичных разрядов в мантиссе данной вычислительной машины, r — число итераций в методе Якоби, n — порядок

матрицы, для которой находятся собственные значения и собственные векторы.

Если в рассмотренной вероятностной оценке возьмем $t = 24$ (это соответствует разрядности мантиссы ЭВМ ЕС), $n = 10^4$ и $r = 9$ (как утверждается в [97], этого числа итераций вполне достаточно), то получаем следующую оценку близости машинного решения к математическому решению задачи:

$$\left[\frac{\sum (\mu_i - \lambda_i)^2}{\sum \lambda_i^2} \right]^{1/2} \leq 10^{-3}.$$

На одно вращение в этом методе требуются две операции деления, $(4n + 28)$ операций умножения, $(2n + 12)$ операций сложения. Справедлива следующая оценка числа итераций:

$$r \leq \frac{\ln e - \ln (\max_{i,j} |a_{ij}|^2 n^2)}{\ln \left(1 - \frac{2}{n(n-1)} \right)}.$$

При вычислении собственных значений и векторов матрицы в памяти ЭВМ требуется хранить $\frac{n^2 + n}{2} + n^2$ чисел, при вычислении только собственных значений $\frac{n^2 + n}{2}$ чисел.

3. Метод Хаусхолдера приведения симметричной матрицы к трехдиагональному виду. Пусть необходимо вычислить собственные значения симметричной матрицы A . Эта задача может быть решена в два этапа: сначала осуществляется приведение исходной матрицы к трехдиагональному виду одним из прямых методов; затем определяются собственные значения симметричной трехдиагональной матрицы. Так как собственные значения симметричной трехдиагональной матрицы определяются сравнительно просто (см., например, метод половинного деления (в п. 1 параграфа III.4) или QL-метод, п. 5 настоящего параграфа), то использование комбинации прямых методов с методами, позволяющими определить собственные значения трехдиагональной матрицы, является весьма эффективным средством определения собственных значений симметричной матрицы.

Метод Хаусхолдера приведения симметричной матрицы $A = A_1$ к трехдиагональной форме состоит из $n - 2$ шагов

$$A_{r+1} = Q_r A Q_r, \quad r = 1, 2, \dots, n-2,$$

где Q_r — матрицы отражения.

На r -м шаге получаются нули в $(n - r + 1)$ -й строке и в $(n - r + 1)$ -м столбце, причем сохраняются нули, полученные на предыдущих шагах. Структура элементов перед r -шагом иллюстрируется рис. 5 для случая $n = 7, r = 4$.

Приведем соотношения, которые определяют алгоритм:

$$Q_r = E - \frac{u_r u_r^T}{h_r}, \quad h_r = \frac{1}{2} (u_r^T u_r),$$

$$\left[\begin{array}{ccc|c|cc} \times & \times & \times & \times & & \\ \times & \times & \times & \times & & \\ \times & \times & \times & \times & & \\ \hline \times & \times & \times & \times & \times & \\ & & & & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{array} \right]$$

Рис. 5

где

$$\mathbf{u}_r^T = [a_{l,1}^{(r)}, a_{l,2}^{(r)}, \dots, a_{l,l-2}^{(r)}, a_{l,l-1}^{(r)} \pm \sigma_r^{1/2}, 0, \dots, 0],$$

$$l = n - r + 1,$$

$$\sigma_r = (a_{l,l}^{(r)})^2 + (a_{l,l-1}^{(r)})^2 + \dots + (a_{l,l-1}^{(r)})^2.$$

Знак перед σ_r в выражении для $\mathbf{u}_r^T$ следует выбирать таким же, как и знак $a_{l,l-1}^{(r)}$.

Если ввести обозначения

$$\mathbf{p}_r = A_r \mathbf{u}_r / h_r, \quad K_r = \mathbf{u}_r^T \mathbf{p}_r,$$

$$\mathbf{q}_r = \mathbf{p}_r - K_r \mathbf{u}_r,$$

то r -й шаг алгоритма (III.34) запишется в виде, удобном для численной реализации

$$A_{r+1} = A_r - \mathbf{u}_r \mathbf{q}_r^T - \mathbf{q}_r \mathbf{u}_r^T.$$

Матрица A_{r+1} имеет трехдиагональный вид для последних r строк и столбцов. Через $(n-2)$ шага получим матрицу A_{n-1} , ортогонально подобную A .

Если $\mathbf{z}$ — собственный вектор трехдиагональной матрицы A_{n-1} , то $Q_1 Q_2 \dots Q_{n-2} \mathbf{z}$ — собственный вектор матрицы A .

Пример 3. Привести к трехдиагональному виду матрицу

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

методом Хаусхолдера.

Решение заключается в вычислении одного преобразования отражения:

$$\mathbf{u}_1^T = [1, 2 \div \sqrt{5}, 0], \quad h_1 = 5 \div 2 \sqrt{5},$$

$$Q_1 = \frac{1}{5 \div 2 \sqrt{5}} \begin{bmatrix} 4 \div 2 \sqrt{5} & -(2 \div \sqrt{5}) & 0 \\ -(2 \div \sqrt{5}) & -(4 \div 2 \sqrt{5}) & 0 \\ 0 & 0 & -5 \div 2 \sqrt{5} \end{bmatrix}.$$

$$A_2 = Q_1 A_1 Q_1 = \begin{bmatrix} -\frac{1}{5} & -\frac{2}{5} & 0 \\ -\frac{2}{5} & \frac{21}{5} & \sqrt{5} \\ 0 & \sqrt{5} & 1 \end{bmatrix}.$$

Для выполнения преобразования отражения не обязательно иметь элементы матрицы Q в явном виде, поскольку она полностью определяется вектором $\mathbf{u}^T$. Метод Хаусхолдера для своей реализации требует

порядка $\frac{2}{3} n^3$ операций умножения и $(n-2)$ извлечений квадратного корня.

Вычисленная по методу Хаусхолдера трехдиагональная матрица A_{n-1} ортогонально подобна матрице $A + dA$, где $\|dA\|_E \leq K n^2 2^{-t} \|A\|_E$, K — константа, зависящая от способа округления, t — число двоичных знаков мантииссы числа.

Отметим, что если матрица имеет кратные корни, то трехдиагональная матрица имеет в случае точных вычислений нулевые внедиагональные элементы. В то же время наличие нулевого внедиагонального элемента в трехдиагональной матрице еще не свидетельствует о том, что имеются кратные корни.

4. Методы Гивенса, Шварца приведения симметричной матрицы к трехдиагональному виду.

Приведение симметричной матрицы к трехдиагональному виду может быть осуществлено с помощью преобразований вращения и описывается следующим соотношением:

$$A_{r+1} = P_r^T A_r P_r, \quad r = 1, 2, \dots,$$

где A_1 — исходная матрица, $P_r = P_{ij}$ — элементарная матрица вращения

$$P_{ij} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & c & \dots & s \\ & & & \vdots & \ddots & \vdots \\ & & & \vdots & \ddots & \vdots \\ & & & -s & \dots & c \\ & & & & \ddots & \\ & & & & & \ddots \\ & & & & & & 1 \end{bmatrix}.$$

При умножении A справа на P_{ij} изменяются только столбцы матрицы A с номерами i и j , элементы которых вычисляются по формулам

$$a_{ki}^+ = ca_{ki} + sa_{kj},$$

$$a_{kj}^+ = -sa_{ki} + ca_{kj}, \quad k = 1, 2, \dots, n.$$

При умножении слева на P_{ij}^T изменяются только строки с номерами i и j , которые вычисляются по формулам

$$a_{ik}^- = ca_{ik} + sa_{jk},$$

$$a_{jk}^- = -sa_{ik} + ca_{jk}, \quad k = 1, 2, \dots, n.$$

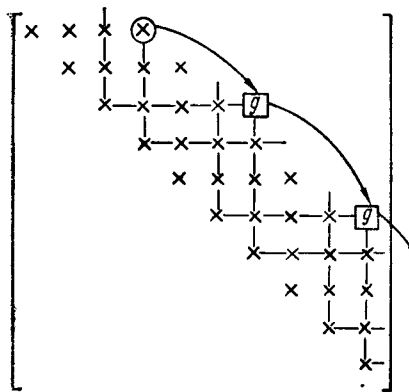


Рис. 6

Тот или иной алгоритм триангуляции определяется выбором чисел c и s .

В методе Гивенса числа c и s подбираются так, чтобы элемент a_{ij} аннулировался при умножении матрицы A справа на матрицу $P_{i+1,j}$. Это произойдет, например, при значениях

$$c = \frac{a_{i,i+1}}{\sqrt{a_{i,i+1}^2 + a_{i,j}^2}},$$

$$s = \frac{a_{ij}}{\sqrt{a_{i,i+1}^2 + a_{ij}^2}},$$

$$i = 1, 2, \dots, n-2.$$

Чтобы сохранить ранее образованные нули, выбирается следующий порядок аннулирования элементов матрицы. Начиная с элемента a_{13} аннулируем все элементы $a_{1,j}$, $j \geq 3$, первой строки. При этом каждый раз преобразуются столбцы с номерами 2 и j . Затем производим аннулирование элементов второй строки, начиная с элемента a_{24} . Продолжая таким же образом аннулирование элементов последующих строк, получаем трехдиагональную матрицу $C = P^*AP$, подобную A .

Для полного приведения симметричной матрицы к трехдиагональному виду требуется порядка $\frac{4}{3}n^3$ умножений и $\frac{1}{2}n^3$ извлечений квадратного корня.

В алгоритме Шварца [187] приведения ленточной симметричной матрицы к трехдиагональному виду числа c и s выбираются так, чтобы исключение необходимого элемента ленточной матрицы приводило к появлению только двух одинаковых в общем случае ненулевых элементов $\tilde{a}$ с симметричным расположением вне полосы исходной матрицы. Для сохранения ленточной структуры матрицы эти элементы исключаются дополнительными вращениями, обеспечивающими сдвиг на m столбцов и на m строк вниз в конечном счете за пределы матрицы.

Для матрицы с параметрами $n = 10$, $m = 3$ исключение начинается с элемента a_{14} . Схема процесса исключения его показана на рис. 6. Матрица симметрична, поэтому на рисунке изображены лишь элементы, находящиеся на главной диагонали и выше.

Умножение исходной матрицы на $P_{3,4}$ исключает элемент a_{14} , но воспроизводит элемент a_{37} . Этот элемент исключается с помощью умножения на матрицу $P_{6,7}$. Но в этом случае появляется элемент $\tilde{a}_{6,10}$. Преобразование $P_{9,10}$ исключает элемент $a_{6,10}$ и тем самым в конечном счете полностью исключает элемент a_{14} .

Следующим исключаем элемент a_{13} , начиная с вращения $P_{2,3}$. Два последующих вращения будут смещать вниз и за пределы матрицы появившийся после первого вращения элемент на пересечении второй строки и шестого столбца. Таким образом, первая строка (столбец)

окажется приведенной к нужной форме. Очевидно, что продолжение процесса над элементами следующих строк не будет изменять уже исключенных элементов и в конечном счете преобразует матрицу к трехдиагональной форме за конечное число вращений.

Следует отметить, что для всех матриц вращения их индексы различаются на единицу и, таким образом, при выполнении одного вращения изменяются лишь две соседних строки (два соседних столбца) матрицы A_k .

Выпишем формулы для элементов матриц $P_{i,j}$, осуществляющих описанный алгоритм. Для исключения элементов $a_{j,j+k}$, $j = 1, 2, \dots, n-2$, $k = m, m-1, \dots, 2$, элементы матриц вращения $P_{j+k-1,j+k}$ определяются по следующим формулам:

$$x = \sqrt{a_{j,j+k}^2 + a_{j,j+k-1}^2}, \quad s = \frac{a_{j,j+k}}{x}, \quad c = \frac{a_{j,j+k-1}}{x}.$$

Для каждого $j = 1, 2, \dots, n-2$, $k = m, m-1, \dots, 2$ выполняются l дополнительных вращений для аннулирования появившихся элементов. Элементы матриц вращений

$$P_{j+k+lm-1,j+k+lm}, \quad l = 1, 2, \dots, \left[\frac{n-k-j}{m} \right]$$

в этом случае будут иметь вид

$$x = \sqrt{\tilde{a}_{j+k+(l-1)m-1,j+k+lm}^2 + \tilde{a}_{j+k+(l-1)m-1,j+k+lm-1}^2},$$

$$s = \frac{\tilde{a}_{j+k+(l-1)m-1,j+k+lm-1}}{x}, \quad c = \frac{\tilde{a}_{j+k+(l-1)m-1,j+k+lm}}{x}.$$

Символами с тильдой обозначены вновь появившиеся элементы вне ленты исходной матрицы после каждого умножения на элементарную матрицу вращения.

Для оценки общего числа необходимых вращений N_r можно указать верхнюю границу, не учитывая сокращение длины верхних диагоналей, и тогда

$$N_r \leq \frac{n^2(m-1)}{2m}.$$

Каждое вращение предполагает одну операцию извлечения квадратного корня и $(8m+13)$ операций умножения, тогда полное число операций умножения равно $n^2(m-1)\left(4 + \frac{6,5}{m}\right)$. Отсюда следует, что стандартный способ приведения симметричной матрицы к трехдиагональному виду методом Хаусхолдера (см. п. 3 настоящего параграфа) будет более быстрым при $m > n/6$. Однако использование ленточного алгоритма диктуется обычно соображениями, связанными с памятью ЭВМ.

После приведения ленточной матрицы к трехдиагональному виду целесообразно использовать QL-метод, если необходимо вычислить все собственные значения матрицы, или метод деления отрезка пополам в случае вычисления отдельных собственных значений.

5. **QR- и QL-методы.** В основу QR-алгоритма положено ортогональное приведение исходной матрицы к треугольной, так что

$$A = QR, \quad (\text{III.37})$$

где Q — ортогональная матрица, R — верхняя треугольная матрица, и, следовательно, матрица

$$B = RQ = Q^T A Q, \quad (\text{III.38})$$

ортогонально подобна матрице A . Таким образом, можно сформировать последовательность ортогонально подобных матриц согласно соотношениям

$$A_k = Q_k R_k, \quad A_{k+1} = R_k Q_k = Q_k^T A_k Q_k, \quad (\text{III.39})$$

предел которой — диагональная матрица, если исходная матрица симметричная, и верхняя квазистреугольная, если исходная матрица несимметричная.

Аналогично определяются QL-преобразования с учетом того, что в формулах (III.37), (III.38), (III.39) вместо R необходимо принять матрицу L , являющуюся нижней треугольной матрицей. В этом случае пределом последовательности матриц A_s будет диагональная для симметричных матриц и нижняя квазистреугольная для несимметричных.

Для любых квадратных матриц существуют QL- и QR-преобразования, поэтому, излагая общую схему алгоритмов, мы не различаем симметричные и несимметричные матрицы. Как видно, QL-алгоритм представляет собой простую реорганизацию QR-алгоритма, и в теоретическом плане оба метода различать не стоит.

Скорость сходимости алгоритма, определенного формулой (III.39), в принципе недостаточна. Сходимость может быть улучшена, если на каждом шаге вместо матрицы A_k использовать матрицу $A_s = k_s E$ (алгоритм со сдвигами) [191]. Последовательность вычислений в этом случае описывается следующими соотношениями:

$$A_s - k_s E = Q_s L_s, \quad A_{s+1} = L_s Q_s + k_s,$$

которые определяют матрицу

$$A_{s+1} = Q_s^T A_s Q_s,$$

по-прежнему ортогонально подобную A .

Если исходная матрица A — симметричная, то все матрицы A_s симметричные, если A — ленточная, то все A_s ленточные. В случае, когда матрица A симметричная и трехдиагональная, процесс особенно прост, поэтому симметричную матрицу целесообразно привести к трехдиагональной с помощью метода Хаусхолдера или Шварца (для ленточных матриц).

Для несимметричных матриц общий подход аналогичен: сначала исходную матрицу целесообразно привести методом Хаусхолдера к форме Хессенберга, затем применить QR- или QL-алгоритм со сдвигом.

6. **QL-метод нахождения собственных значений симметричной трехдиагональной матрицы** сводится к преобразованию в ходе итерационного процесса исходной трехдиагональной матрицы A к диагональной с помощью преобразований подобия

$$A_{k+1} = Q_k A_k Q_k^T = Q_k Q_{k-1} A_{k-1} Q_{k-1}^T Q_k^T = Q_k \dots Q_1 A_1 Q_1^T \dots Q_k^T.$$

Здесь $A_1 = A$.

По существу, QL-метод реализуется следующим образом:

$$Q_k A_k = L_k, \quad A_{k+1} = L_k Q_k^T, \quad k = 1, 2, \dots$$

где Q_k — ортогональная матрица, преобразующая A_k в левую треугольную матрицу L_k , k — номер итерации. При достаточно больших k матрица A_k становится близкой к диагональной. Диагональные элементы ее являются приближенными собственными значениями матрицы A .

Матрица преобразования Q_k на каждом шаге итерации является произведением элементарных матриц вращения (см. п. 4 параграфа III.1) или отражения (см. п. 5 параграфа III.1).

Рассмотрим вариант метода, основанный на использовании матриц вращения. Пусть задана матрица

$$A_k = \begin{bmatrix} d_1^{(k)} & l_1^{(k)} & 0 & \dots & 0 & 0 & 0 \\ l_1^{(k)} & d_2^{(k)} & l_2^{(k)} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & l_{n-2}^{(k)} & d_{n-1}^{(k)} & l_{n-1}^{(k)} \\ 0 & 0 & 0 & \dots & 0 & l_{n-1}^{(k)} & d_n^{(k)} \end{bmatrix}. \quad (\text{III.40})$$

Аннулируем элемент $l_{n-1}^{(k)}$, используя матрицу вращения, которая имеет вид

$$P_{n-1}^{(k)} = \begin{bmatrix} 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 1 & 0 & 0 \\ 0 & \dots & 0 & c & -s \\ 0 & \dots & 0 & s & c \end{bmatrix}.$$

В результате преобразования $P_{n-1}^{(k)} A_k$ аннулируется элемент $l_{n-1}^{(k)}$, а преобразование $P_{n-1}^{(k)} A_k (P_{n-1}^{(k)})^T$ приводит, с одной стороны, к появлению нового элемента $l_{n-1}^{(k)}$, меньшего по сравнению с предыдущим своим значением, с другой — к появлению на месте $(n-2, n)$, и, следовательно, в силу симметрии матрицы на месте $(n, n-2)$ ненулевого элемента $g_1^{(k)}$. На схеме (см. рис. 7) элементы матрицы A_k обозначены значком $\times$, вновь появившиеся элементы $(n, n-1)$ и $(n-1, n)$ — точками, а элемент $g_1^{(k)}$ — значком $\otimes$.

Таким образом, на данном этапе вычислений матрица A_k перестает быть трехдиагональной, поэтому ее необходимо привести к трехдиа-

$$\left[\begin{array}{cccc} \times & \times & & \\ \times & \times & \times & \\ & \times & \times & \times \\ & & \times & \times & \times & \otimes \\ & & & \times & \times & \cdot \\ & & & & \times & \times & \cdot \\ & & \otimes & \cdot & \times & & \end{array} \right] \quad \left[\begin{array}{cccc} \times & \times & & \\ \times & \times & \times & \\ & \times & \times & \times & \otimes \\ & & \times & \times & \times & 0 \\ & & & \times & \times & \times & \cdot \\ & & \otimes & \times & \times & \cdot \\ & & & 0 & \cdot & \times \end{array} \right]$$

Рис. 7

Рис. 8

гональной. Для этого на данном шаге вычисляют коэффициенты вращения

$$c_{n-1}^{(k)} = \frac{l_{n-2}^{(k)}}{\sqrt{(l_{n-2}^{(k)})^2 + (g_1^{(k)})^2}}, \quad s_{n-1}^{(k)} = \frac{g_1^{(k)}}{\sqrt{(l_{n-2}^{(k)})^2 + (g_1^{(k)})^2}}$$

и преобразуют строки и столбцы с номерами $n-1$ и $n-2$ по формулам метода Гивенса (см. п. 4 настоящего параграфа).

В результате g_1 становится равным нулю, а на месте элементов с индексами $(n-3, n-1)$ и, следовательно, $(n-1, n-3)$ появляется элемент $g_2^{(k)}$ (рис. 8).

Для аннулирования вновь возникающего элемента снова применяют формулы Гивенса и т. д. Только аннулирование элемента $g_{n-2}^{(k)}$, имеющего индексы $(1, 3)$, $(3, 1)$, не вызывает возникновения элемента g . На этом шаг процесса считают законченным. Шаги итерационного процесса повторяют до тех пор, пока элемент $l_{n-1}^{(k)}$ не станет достаточно малым. В этом случае элемент с индексами (n, n) может быть принят за приближение к собственному значению матрицы A . После получения приближенного собственного значения рассматривают матрицу, полученную из исходной вычеркиваниями последнего столбца и последней строки, порядок которой на единицу меньше предыдущей, и повторяют вновь всю итерационную процедуру вычисления следующего собственного значения. И так продолжают до тех пор, пока не вычислят все собственные значения с заданной степенью точности. Таким образом, вместо трехдиагональной матрицы получают приближенно-диагональную.

Пример 4. Пусть требуется найти собственные значения матрицы

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

QL-методом.

Решение. Аннулируем элемент a_{23} . Для этого вычисляем s и c в матрице вращения P_2 :

$$s = \frac{a_{23}}{\sqrt{a_{23}^2 + a_{33}^2}} = -0,4472, \quad c = \frac{a_{33}}{\sqrt{a_{23}^2 + a_{33}^2}} = 0,8944.$$

Нетрудно убедиться, что

$$P_2 A P_2^{-1} = \begin{bmatrix} 2,0000 & -0,8944 & 0,4472 \\ -0,8944 & -1,2000 & -0,6000 \\ 0,4472 & -0,6000 & 2,8000 \end{bmatrix}.$$

Теперь аннулируем элемент a_{13} матрицы $P_2 A P_2^{-1}$. Для этого вычисляем $s = 0,5976$; $c = -0,8018$. После проведения соответствующего преобразования подобия получим матрицу

$$\begin{bmatrix} 0,8571 & -0,6389 & 0 \\ -0,6388 & 2,3428 & 0,7483 \\ 0 & 0,7483 & 2,8000 \end{bmatrix}.$$

Снова аннулируем элемент a_{23} , затем a_{13} и т. д. На двенадцатой итерации получаем

$$\begin{bmatrix} 1,7798 & 0,5127 & 0 \\ 0,5127 & 0,8059 & 0,0003 \\ 0 & 0,0003 & 3,4142 \end{bmatrix}.$$

Из рассмотрения последней матрицы видно, что с точностью до трех десятичных знаков нами получено собственное значение

$$\lambda = 3,4142.$$

Можно продолжить вычисления дальше, работая с матрицей меньшего порядка на единицу, в данном случае с матрицей второго порядка. Для этой матрицы собственные числа вычисляются как корни полинома

$$\det \begin{bmatrix} 1,7798 - \lambda & 0,5127 \\ 0,5127 & 0,8059 - \lambda \end{bmatrix} = 0.$$

Нетрудно убедиться, решив квадратное уравнение, что $\lambda_1 = 0,5858$, $\lambda_2 = 2,0000$. Все вычисления в этом примере проводились с точностью до четырех знаков после запятой.

В этом методе справедлива следующая оценка близости машинного и математического решения задачи

$$\frac{\|\mu_i - \lambda_i\|}{\|A\|_E} \leq Cr 2^{-t},$$

где C — некоторая константа, r — число итераций, t — число двоичных разрядов мантиссы на данной машине, μ_i — собственное значение матрицы, полученное на машине, λ_i — точное собственное значение исходной матрицы.

На практике обычно применяют так называемый QL-метод со сдвигами, а именно:

$$Q_k(A_k - q_k E) = L_k, \quad A_{k+1} = L_k Q_k^T + q_k E.$$

Здесь q_k тот из корней квадратного уравнения

$$\det \begin{bmatrix} d_{n-1}^{(k)} - \mu_k & l_{n-1}^{(k)} \\ l_{n-1}^{(k)} & d_n^{(k)} - \mu_k \end{bmatrix} = 0, \quad (\text{III.41})$$

который ближе к $d_n^{(k)}$.

По сравнению с QL -методом без сдвигов QL -метод со сдвигами позволяет получить решение на ЭВМ с той же точностью за меньшее число итераций (следовательно, арифметических операций).

Вычислительная схема QL -метода со сдвигами может быть реализована следующим образом. Вначале решаем уравнение (III.41),

$$\mu_k^2 - (d_n^{(k)} + d_{n-1}^{(k)})\mu_k + d_{n-1}^{(k)}d_n^{(k)} - (l_{n-1}^{(k)})^2 = 0,$$

по формулам

$$\mu_k^{(1,2)} = \frac{d_n^{(k)} + d_{n-1}^{(k)}}{2} \pm \sqrt{\frac{(d_n^{(k)} + d_{n-1}^{(k)})^2}{4} + (l_{n-1}^{(k)})^2 - d_{n-1}^{(k)}d_n^{(k)}}$$

и определяем коэффициент сдвига q_k . Далее вычислив коэффициенты вращения

$$c_{n-1}^{(k)} = \frac{d_n^{(k)} - q_k}{P_{n-1}^{(k)}}, \quad s_{n-1}^{(k)} = \frac{l_{n-1}^{(k)}}{P_{n-1}^{(k)}}, \quad P_{n-1}^{(k)} = \sqrt{(d_n^{(k)} - q_k)^2 + (l_{n-1}^{(k)})^2},$$

преобразуем n и $(n-1)$ строки (обозначим их $A_k^{(n)}$ и $A_k^{(n-1)}$ соответственно) матрицы A_k по формулам

$$\tilde{A}_k^{(n)} = c_{n-1}^{(k)}A_k^{(n)} + s_{n-1}^{(k)}A_k^{(n-1)},$$

$$\tilde{A}_k^{(n-1)} = -s_{n-1}^{(k)}A_k^{(n)} + c_{n-1}^{(k)}A_k^{(n-1)}.$$

При этом элемент $l_{n-1}^{(k)}$ становится равным нулю. Преобразуем n и $(n-1)$ столбцы по тем же формулам, что и для строк. Теперь элемент $l_{n-1}^{(k)}$ стал ненулевым, но меньшим по модулю своего предыдущего значения, но на местах $(n-2, n)$ и $(n, n-2)$ появляется ненулевой элемент g_1 (см. рис. 7), и обрабатываемая матрица перестает быть трехдиагональной.

Теперь приведем полученную матрицу методом Гивенса к трехдиагональному виду за $n-2$ шага.

Для этого вычислим коэффициенты вращения метода Гивенса

$$c_{n-1}^{(k)} = \frac{l_{n-2}^{(k)}}{P_{n-1}^{(k)}}, \quad s_{n-1}^{(k)} = \frac{g_{n-2}^{(k)}}{P_{n-1}^{(k)}}, \quad P_{n-1}^{(k)} = \sqrt{(l_{n-2}^{(k)})^2 + g_{n-2}^{(k)2}}$$

и преобразуем $(n-2)$ и $(n-1)$ строки и столбцы матрицы. В результате значение g_1 становится равным нулю, но появляется элемент g_2 на местах $(n-3, n-1)$ и $(n-1, n-3)$ (см. рис. 8). Повторяем шаг этой процедуры для элементов с индексами $(n-3, n-1)$ и т. д. Аннулирование элемента g_{n-2} , отмеченного на рис. 9 значком $\otimes$, не вызывает возникновения элемента g . На этом шаг QL -метода заканчива-

ется. Процесс повторяется снова для аннулирования элементов с индексами $(n-1, n)$ и $(n, n-1)$, до тех пор пока эти элементы не станут достаточно малыми. При этом элемент с индексами (n, n) принимают за приближенное собственное значение исходной матрицы.

Чтобы найти следующее собственное значение, вычеркивают последний столбец и строку и работают с матрицей, порядок которой на единицу меньше предыдущей.

В сочетании с методом Хаусхолдера или Гивенса (ленточный)

QL -метод является одним из эффективных методов вычисления собственных значений симметричной матрицы. Высокая устойчивость, быстрая сходимость итераций, небольшой объем требуемой машинной памяти (ограничения на память накладываются лишь методом Хаусхолдера или Гивенса) делают это сочетание предпочтительным для машинной реализации.

7. Приведение матриц общего вида к форме Хессенберга. Для вычисления собственных значений и собственных векторов несимметричных матриц вначале посредством ортогональных преобразований приводят исходную матрицу к форме Хессенберга. Затем матрицу Хессенберга с помощью QR -метода (см. п. 8 настоящего параграфа) сводят к квазитреугольному виду. В квазитреугольной матрице клетки на диагонали имеют собственные значения, близкие к собственным значениям исходной матрицы.

Рассмотрим приведение действительной матрицы

$$A_1 = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

к верхней форме Хессенберга

$$\tilde{A} = \begin{bmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \dots & \tilde{a}_{1,n-1} & \tilde{a}_{1,n} \\ \tilde{a}_{21} & \tilde{a}_{22} & \dots & \tilde{a}_{2,n-1} & \tilde{a}_{2,n} \\ 0 & \tilde{a}_{32} & \dots & \tilde{a}_{3,n-1} & \tilde{a}_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \tilde{a}_{n,n-1} & \tilde{a}_{n,n} \end{bmatrix}$$

посредством элементарных ортогональных преобразований отражения (см. п. 5 параграфа III.1).

Приведение исходной матрицы к форме Хессенберга осуществляется за $(n-2)$ шага. Матрица A_{r+1} связана с матрицей A_r следующим

$$\begin{bmatrix} \times & \times & \otimes & & \\ \times & \times & \times & & \\ \otimes & \times & \times & \times & \\ & \times & \times & \times & \\ & & \times & \times & \epsilon \\ & & & \epsilon & \times \end{bmatrix}$$

Рис. 9

преобразованием:

$$A_{r+1} = Q_r A_r Q_r^T, \quad r = 1, 2, \dots, n-2. \quad (\text{III.42})$$

Здесь Q_r — матрица отражения,

$$Q_r = E - \frac{u_r u_r^T}{h_r}, \quad (\text{III.43})$$

где

$$h_r = \frac{1}{2} (u_r^T u_r),$$

$$\sigma_r = (a_{r+1,r}^{(r)})^2 + (a_{r+2,r}^{(r)})^2 + \dots + (a_{n,r}^{(r)})^2, \quad (\text{III.44})$$

$$u_r^T = [0, \dots, 0, a_{r+1,r}^{(r)} \pm \sigma_r^{1/2}, a_{r+2,r}^{(r)}, \dots, a_{n,r}^{(r)}].$$

Очевидно, что нулевые элементы, образованные на предыдущих шагах в первых $(r-1)$ столбцах, остаются без изменений.

Матрицу A_{r+1} удобно вычислять в два этапа: сначала сформировать матрицу

$$B_{r+1} = Q_r A_r = A_r - \frac{u_r (u_r^T A_r)}{h_r},$$

а затем —

$$A_{r+1} = B_{r+1} Q_r = B_{r+1} - \frac{B_{r+1} u_r}{h_r} u_r^T.$$

При вычислении с t разрядной мантиссой полученная матрица A_{n-1} ортогонально подобна некоторой матрице $A_1 + dA$, где $\|dA\|_E \leq K_1 n^{2-t} \|A_1\|_E$, K_1 — константа порядка единицы.

Приведение к форме Хессенберга также может быть осуществлено методом Гивенса с помощью элементарных устойчивых преобразований вращения.

Структуру ленточных матриц сохраняет алгоритм Шварца (см. п. 4 настоящего параграфа).

На практике перед приведением к форме Хессенберга целесообразно сбалансировать матрицу (см. [98]), преобразуя ее к виду

$$B = D M A M^{-1} D^{-1},$$

где M — матрица перестановок, а D — диагональная матрица с положительными элементами. Матрица B имеет вид

$$B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ 0 & B_{22} & B_{23} \\ 0 & 0 & B_{33} \end{bmatrix}.$$

Матрицы D и M выбирают таким образом, чтобы B_{11} и B_{33} были верхними треугольными, а матрица B_{22} — равновесной, т. е. чтобы ее строки имели приблизительно одинаковые l_1 -нормы. Таким образом, собственные значения матрицы A будут диагональными элементами матриц B_{11} и B_{33} и собственные значения матрицы B_{22} . Если матрица B_{22} объ-

единяет строки и столбцы с номерами от k до l , то очевидно, что приведение исходной матрицы к форме Хессенберга включает в себя лишь действия над этими строками и столбцами.

8. QR-метод для вычисления собственных значений матрицы Хессенберга. Матрица Хессенберга может быть приведена к квазитреугольному виду с помощью QR-метода. Этот метод описывается следующими соотношениями:

$$Q_k A_k = R_k, \quad A_{k+1} = R_k Q_k^T, \quad k = 1, 2, \dots, \quad (\text{III.45})$$

или

$$A_{k+1} = Q_k A_k Q_k^T,$$

где Q — ортогональная, R — верхняя треугольная матрица. При этом A_k подобна исходной, так как $Q^T = Q^{-1}$. В процессе итераций матрицы A_k сходятся по форме к верхней квазитреугольной матрице, клетки на диагонали которой имеют собственные значения, близкие к собственным значениям исходной матрицы. Например, паре комплексных собственных значений $p \pm iq$ матрицы A соответствует диагональная клетка второго порядка квазитреугольной матрицы. Собственному числу матрицы A кратности θ соответствует диагональная клетка порядка θ квазитреугольной матрицы.

Целесообразно использовать QR-метод с восстановлением сдвига; этот вариант работает быстрее обычного QR-метода. Алгоритм со сдвигами заключается в следующем. Вычисляется последовательность матриц

$$A_s - k_1 E = Q_s R_s, \quad R_s Q_s + k_1 E = A_{s+1},$$

$$A_{s+1} - k_2 E = Q_{s+1} R_{s+1}, \quad R_{s+1} Q_{s+1} + k_2 E = A_{s+2},$$

где $s = 1, 2, \dots$; Q_s — ортогональные матрицы; R_s — правые треугольные матрицы; k_1, k_2 — собственные значения матрицы второго порядка правого нижнего угла матрицы A_s .

Во избежание комплексности при вычислении k_1 и k_2 два шага процесса объединены в один, в результате чего происходит переход от матрицы A_s к матрице A_{s+2} . Для этого вычисляются величины

$$x = a_{11}^2 + a_{12}a_{21} - a_{11}(a_{n-1,n-1} + a_{nn}) + a_{n-1,n-1}a_{nn} - a_{n-1,n}a_{n,n-1},$$

$$y = a_{21}(a_{11} + a_{22} - a_{n-1,n-1} - a_{nn}), \quad z = a_{32}a_{21},$$

$$c_1 = \frac{x}{\sqrt{x^2 + y^2}}, \quad s_1 = \frac{y}{\sqrt{x^2 + y^2}}.$$

Преобразуются первый и второй столбцы по формулам

$$\left. \begin{aligned} a_{i1} &= ca_{i1} + sa_{i2} \\ a_{i2} &= -sa_{i1} + ca_{i2} \end{aligned} \right\} \quad i = 1, 2, 3$$

и первая и вторая строки по формулам

$$\left. \begin{aligned} a_{1i} &= ca_{1i} + sa_{2i} \\ a_{2i} &= -sa_{1i} + ca_{2i} \end{aligned} \right\} \quad i = 1, 2, \dots, n.$$

Затем вычисляются величины

$$c_2 = \frac{x}{\sqrt{x^2 + z^2}}, \quad s_2 = \frac{z}{\sqrt{x^2 + z^2}}$$

и по формулам, аналогичным приведенным выше для $i = 1, 2, 3, 4$, преобразуются первый и третий столбцы и соответствующие строки полученной матрицы. Будем иметь матрицу

$$C = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1,n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2,n-1} & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3,n-1} & a_{3n} \\ a_{41} & 0 & a_{43} & \dots & a_{4,n-1} & a_{4n} \\ 0 & 0 & a_{53} & \dots & a_{5,n-1} & a_{5n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{n,n-1} & a_{nn} \end{bmatrix}.$$

Матрица C с учетом ее специфики приводится методом Гивенса к виду Хессенберга. В результате получается матрица A_{s+2} , подобная A_1 . Процесс продолжается до тех пор, пока матрица A_s не примет квазитреугольную форму.

Итерационный процесс QR -метода сводит матрицу Хессенберга A_1 к квазитреугольной форме A_s , клетки по диагонали которой имеют собственные числа, близкие к собственным значениям исходной матрицы. Анализ ошибок машинной реализации показывает, что машинная матрица будет точно ортогонально подобна некоторой матрице $A_1 + dA_1$, причем справедлива оценка

$$\|dA_1\|_E \leq K_2 2^{-t} ns \|A_1\|_E,$$

где A_1 — исходная матрица Хессенберга, K_2 — константа порядка единицы, s — полное количество итераций, n — порядок матрицы, t — число двоичных разрядов мантиисы на данной машине. В п. 2 и 3 параграфа III.2 для различных собственных значений λ_i матрицы A приведены следующие оценки зависимости возмущений собственных чисел и собственных векторов от возмущения коэффициентов исходной матрицы ΔA :

$$|\Delta \lambda_i| \leq \text{cond } \lambda_i \|\Delta A\|_E,$$

$$\frac{\|\Delta v_i\|}{\|v_i\|} \leq \|\Delta A\|_E \sum_{j, j \neq i} \frac{\text{cond } \lambda_j}{|\lambda_i - \lambda_j|}.$$

Трактуя ошибки вычислений метода, как возмущение элементов исходной матрицы, получаем оценку близости машинного и математического решения

$$|d\lambda_i| \leq \text{cond } \lambda_i 2^{-t} k_2 ns \|A\|_E,$$

$$\frac{\|dv_i\|}{\|v_i\|} \leq 2^{-t} k_2 ns \|A\|_E \sum_{j, j \neq i} \frac{\text{cond } \lambda_j}{|\lambda_i - \lambda_j|}.$$

Величина

$$\log_{10} \left(\frac{|\lambda_i|}{\text{cond } \lambda_i \|dA\|_E} \right)$$

будет гарантировать количество верных знаков вычисленного собственного числа λ_i [137] в случае, когда величина $\|dA\|_E$ незначительна.

Если для определения собственных значений несимметричной действительной матрицы используется метод Хаусхолдера в сочетании с QR -алгоритмом, то значение $\text{cond } \lambda$ может быть вычислено по алгоритму, имеющемуся в работе [137] без явного вычисления левых и правых собственных векторов исходной матрицы.

Помимо вычислительных трудностей при нахождении собственных значений несимметричных матриц может оказаться, что собственные значения весьма чувствительны к малым изменениям элементов матрицы. У таких матриц могут оказаться нелинейные элементарные делители (см. п. 7 параграфа III.1)¹⁴ и в этом случае собственные векторы не порождают полного пространства. Поэтому при нахождении собственных значений несимметричных матриц целесообразно одновременно находить полное число собственных векторов. Анализируя вычисленные собственные векторы, бывает нетрудно выяснить, что точная матрица имеет дефект¹⁵. В этом случае вычисленные собственные векторы почти параллельны.

III.4. МЕТОДЫ ВЫЧИСЛЕНИЯ НЕСКОЛЬКИХ СОБСТВЕННЫХ ЗНАЧЕНИЙ И ПРИНАДЛЕЖАЩИХ ИМ СОБСТВЕННЫХ ВЕКТОРОВ

1. Метод половинного деления. В некоторых задачах на собственные значения требуется определить несколько или одно из собственных значений или найти собственные значения в заданном интервале. Так, например, при использовании метода верхней релаксации для вычисления итерационного параметра необходимо знать максимальное собственное значение итерированной матрицы простой итерации (см. п. 2 параграфа II. 2). В задачах колебания, сводящихся к задачам на собственные значения, практически представляет интерес несколько первых собственных значений.

Для решения таких задач можно использовать методы нахождения всех собственных значений матриц, рассмотренные выше, но это не выгодно, так как приводит к большим затратам арифметических операций и в конечном итоге времени ЭВМ. Поэтому в данном случае есть смысл применять методы, позволяющие находить несколько собственных значений и более экономичные с точки зрения затрат арифметических операций и времени ЭВМ по сравнению с рассмотренными выше методами.

Одним из таких методов является метод половинного деления, позволяющий находить одно или группу (с заданной последователь-

¹⁴ Нелинейные элементарные делители могут возникать и за счет погрешности машинной реализации.

¹⁵ Следуя [97], будем считать, что матрица имеет дефект, если у матрицы есть один (или больше) нелинейный элементарный делитель.

ностью номеров) собственных значений трехдиагональной матрицы в заданном интервале. Опыт показывает, что этот метод эффективнее QL-метода, если требуется определить не более чем $n/4$ собственных значений (n — порядок симметричной трехдиагональной матрицы).

Пусть задана трехдиагональная матрица

$$A = \begin{bmatrix} a_1 & b_2 & 0 & 0 & \dots & 0 & 0 \\ b_2 & a_2 & b_3 & 0 & \dots & 0 & 0 \\ 0 & b_3 & a_3 & b_4 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & a_{n-1} & b_n \\ 0 & 0 & 0 & 0 & \dots & b_n & a_n \end{bmatrix}.$$

Не нарушая общности рассуждений, можно предположить, что ни один из элементов b_i не равен нулю. Действительно, пусть для некоторого q элемент $b_q = 0$. Тогда можно рассматривать трехдиагональные матрицы: A_1 порядка $q - 1$ и A_2 порядка $n - q + 1$, для них

$$A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}.$$

Собственные значения матриц A_1 и A_2 являются собственными значениями матрицы A . Если v — собственный вектор матрицы A_1 , то соответствующий собственный вектор матрицы A будет

$$[v^T, \underbrace{0, \dots, 0}_{n-q+1}]^T.$$

Можно показать, что у симметричной трехдиагональной матрицы, не имеющей нулевых недиагональных элементов, все собственные значения различны. Если симметричная матрица имеет кратное собственное значение, то трехдиагональная матрица, полученная из нее методом Хаусхолдера, будет иметь несколько ненулевых (или близких к нулю при приближенных вычислениях) недиагональных элементов [97]. Поэтому метод половинного деления можно рассматривать для симметричных трехдиагональных матриц с ненулевыми недиагональными элементами.

Метод половинного деления основывается на рассмотренном выше (см. п. 2 параграфа III.1) свойстве последовательности Штурма. Если интервал (A_0, B_0) , на котором разыскивается собственное значение, неизвестен, то подходящими для A_0 и B_0 являются значения $\pm \|A\|_\infty$. Поскольку матрица имеет трехдиагональный вид, эта норма легко вычисляется:

$$\|A\|_\infty = \max \{|b_i| + |a_i| + |b_{i+1}|\}, \quad b_1 = b_{n+1} = 0.$$

Вычислительная схема метода половинного деления для определения k -го собственного значения может быть реализована следующим образом. До начала вычислений задают значения A_0, B_0 и ϵ (условие окончания итерационного процесса). Затем делят интервал (A_{r-1}, B_{r-1}) (r — номер итерации, $r = 1, 2, \dots$) пополам. Точка $C_r =$

$= \frac{1}{2} (A_{r-1} + B_{r-1})$ является его серединой. Вычисляют последовательность $p_0(C_r), p_1(C_r), \dots, p_n(C_r)$ и определяют число совпадений знаков $s(C_r)$.

Если $s(C_r) \geq k$, то полагают $A_r = C_r, B_r = B_{r-1}$, а в случае $s(C_r) < k$ считают, что $A_r = A_{r-1}, B_r = C_r$. Теперь $\lambda_k \in (A_r, B_r)$. Этот процесс повторяют до тех пор, пока не будет выполнено условие

$$\frac{B_r - A_r}{2} < \epsilon C_{r+1}.$$

При выполнении этого условия значение C_{r+1} принимают в качестве приближенного значения λ_k . Если выполнено m шагов по методу половинного деления, то центр C_{m+1} последнего интервала таков, что

$$|C_{m+1} - \lambda_k| < 15,06 \cdot 2^{-t} + (B_0 - A_0) 2^{-m-1},$$

где t — число двоичных разрядов мантисы числа ЭВМ [97]. Эта оценка носит мажорантный характер и замечательна тем, что не зависит от порядка матрицы.

На каждом шаге метода половинного деления требуются $2n$ умножений и $2n$ вычитаний.

Пример 5. Пусть требуется найти третье собственное значение матрицы

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}$$

методом половинного деления.

Решение. Так как интервал, в котором находится третье собственное значение, неизвестен, определяем $\|A\|_\infty = 4$ и собственные значения будем искать в интервале $(-4, 4)$. Действительно, подставляя значение минус четыре в последовательность Штурма, имеем

$$p_0(-4) = 1, \quad p_1(-4) = 6, \quad p_2(-4) = 35, \quad p_3(-4) = 204, \quad p_4(-4) = 1184.$$

Таким образом, число совпадений знака $s(-4) = 4$. Это означает, что все четыре собственных значения матрицы A больше, чем минус четыре. Подставляем теперь в последовательность Штурма 4, получаем

$$p_0(4) = 1, \quad p_1(4) = -2, \quad p_2(4) = 3, \quad p_3(4) = -4, \quad p_4(4) = 5.$$

Число совпадений знаков $s(4) = 0$ и, значит, все собственные значения матрицы A не больше четырех.

Делим интервал $(-4, 4)$ пополам, получаем

$$C_1 = \frac{1}{2} (-4 + 4) = 0$$

и строим последовательность Штурма для этого значения C_1 :

$$p_0(0) = 1, \quad p_1(0) = 2, \quad p_2(0) = 3, \quad p_3(0) = 4.$$

Так как имеются три совпадения знаков, последующие вычисления значений полиномов можно не выполнять. Таким образом, $s(0) \geq 3$, значит, $\lambda_3 \in (0, 4)$. Далее имеем

$$C_2 = \frac{1}{2} (0 + 4) = 2,$$

$$p_0(2) = 1, \quad p_1(2) = -0, \quad p_2(2) = -1, \quad p_3(2) = +0, \quad p_4(2) = 1.$$

Знак нуля в $p_1(2)$ берется противоположным знаком $p_0(2)$, а знак $p_3(2)$ — противоположным знаком $p_2(2)$. В построенной последовательности Штурма $s(2) = 2 < 3$, значит $\lambda_3 \in (0, 2)$,

$$C_3 = \frac{1}{2} (0 + 2) = 1,$$

$$p_0(1) = 1, \quad p_1(1) = 1, \quad p_2(1) = -0, \quad p_3(1) = -1, \quad p_4(1) = -1.$$

В рассматриваемой последовательности Штурма $s(1) = 3$, значит $\lambda_3 \in (1, 2)$ и т. д. Продолжая этот процесс, получаем $\lambda_3 \approx 1,382$, что хорошо согласуется с истинным значением λ_3 .

2. Степенной метод. Для нахождения максимального по модулю собственного значения и принадлежащего ему собственного вектора квадратной матрицы произвольной структуры можно использовать степенной метод.

Вначале рассмотрим реализацию этого метода для действительной матрицы A простой структуры¹⁶, имеющей собственные значения, которые расположены с учетом их кратности в следующем порядке:

$$|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_{n-1}| < |\lambda_n|.$$

Задавая произвольное начальное приближение y_0 к собственному вектору v_n и значение $\varepsilon > 0$, по которому будет оканчиваться итерационный процесс, степенной метод реализуют по формулам

$$\tilde{y}^{(k)} = A y^{(k-1)}, \quad y^{(k)} = \frac{\tilde{y}^{(k)}}{\max_{1 \leq i \leq n} |\tilde{y}_i^{(k)}|}, \quad \mu_k = \frac{1}{m} \sum_{s=1}^m \frac{\tilde{y}_s^{(k)}}{y_s^{(k-1)}}, \quad (III.46)$$

$$k = 0, 1, \dots,$$

проверяя на каждой итерации условие окончания процесса

$$|\mu_k - \mu_{k-1}| \leq \varepsilon |\mu_k|. \quad (III.47)$$

Отметим, что μ_k можно вычислить также по формуле

$$\mu_k = \max_i |\tilde{y}_i^{(k)}|.$$

Здесь и в дальнейшем мы используем обозначения $\max \{x\}$ для максимального по модулю элемента вектора x .

В (III.46), (III.47) k — номер итерации, s — номер компоненты, m — количество ненулевых компонент вектора y_n , μ_k — приближение к собственному числу λ_n .

Пример 6. Пусть требуется найти максимальное собственное значение матрицы

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (III.48)$$

степенным методом.

¹⁶ Матрицей простой структуры называется матрица A порядка n , имеющая n линейно независимых собственных векторов.

Таблица 4

μ, y_i	k					
	0	1	2	3	4	5
μ	—	5	5,423	5,369	5,372	5,372
y_1	1	0,571	0,595	0,592	0,593	0,593
y_2	1	1	1	1	1	1
y_3	1	0,571	0,595	0,592	0,593	0,593

Решение. Выбрав в качестве начального приближения $y^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, вычислим

$$\tilde{y}^{(1)} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 4 \end{bmatrix}, \quad y^{(1)} = \frac{1}{7} \begin{bmatrix} 4 \\ 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 0,571 \\ 1 \\ 0,571 \end{bmatrix}, \quad \mu_1 = 5.$$

После второй итерации имеем

$$\tilde{y}^{(2)} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0,571 \\ 1 \\ 0,571 \end{bmatrix} = \begin{bmatrix} 3,143 \\ 5,286 \\ 3,143 \end{bmatrix}, \quad y^{(2)} = \begin{bmatrix} 0,595 \\ 1 \\ 0,595 \end{bmatrix},$$

$$\mu_2 = \frac{1}{3} \left(\frac{3,143}{0,571} + \frac{5,286}{1} + \frac{3,143}{0,571} \right) = 5,423.$$

Эти и последующие приближения к собственному вектору, соответствующему наибольшему собственному значению, приведены в табл. 4

Из таблицы видно, что приближение к собственному значению и собственному вектору было получено за пять итераций.

Покажем сходимость степенного метода, когда значение λ_n действительно и не кратно. Так как матрица A есть матрица простой структуры, система собственных векторов $\{v_j\}$, $j = 1, 2, \dots, n$, может быть принята за базис в n -мерном пространстве. Следовательно, произвольный вектор $y^{(0)}$ может быть разложен по собственным векторам

$$y^{(0)} = c_1 v_1 + c_2 v_2 + \dots + c_n v_n.$$

Учитывая тот факт, что собственный вектор определен с точностью до постоянного множителя, при исследовании сходимости не будем принимать во внимание условие нормировки в (III.46).

Из последовательных итераций получаем векторы

$$y^{(1)} = A y^{(0)} = c_1 \lambda_1 y_1 + c_2 \lambda_2 y_2 + \dots + c_n \lambda_n y_n,$$

$$y^{(2)} = A y^{(1)} = c_1 \lambda_1^2 y_1 + c_2 \lambda_2^2 y_2 + \dots + c_n \lambda_n^2 y_n, \quad (III.49)$$

$$y^{(k)} = A y^{(k-1)} = c_1 \lambda_1^k y_1 + c_2 \lambda_2^k y_2 + \dots + c_n \lambda_n^k y_n =$$

$$= \lambda_n^k [c_n y_n + c_{n-1} \left(\frac{\lambda_{n-1}}{\lambda_n} \right)^k y_{n-1} + \dots + c_1 \left(\frac{\lambda_1}{\lambda_n} \right)^k y_1] = \lambda_n^k [c_n y_n + O \left(\frac{\lambda_{n-1}}{\lambda_n} \right)^k].$$

Вычислим отношение соответствующих компонент двух последовательных итераций

$$\frac{y_i^{(k)}}{y_i^{(k-1)}} = \lambda_n + O\left(\frac{\lambda_{n-1}}{\lambda_n}\right)^k. \quad (\text{III.50})$$

Так как λ_n наибольшее по модулю собственное значение, из (III.49) при $c_n \neq 0$ видно, что последовательность итераций степенями матрицы A произвольного вектора $y^{(0)}$ стремится к собственному вектору, принадлежащему максимальному собственному значению. Формула (III.50) показывает, что отношение одноименных компонент векторов двух соседних итераций стремится к максимальному собственному значению.

Из формулы (III.46) видно, что на каждом шаге итерационного процесса приходится нормировать итерированный вектор. Без такой нормировки компоненты итерированного вектора быстро возрастали бы или убывали, и при определенных условиях этот метод был бы не реализуем на ЭВМ. Отметим, что если за нормирующий множитель брать величину, обратную максимальной компоненте вектора, то эта компонента стремится к максимальному собственному значению.

Из формулы (III.50) следует, что скорость сходимости рассматриваемого метода линейна и определяется знаменателем

$$q = \left| \frac{\lambda_{n-1}}{\lambda_n} \right|. \quad (\text{III.51})$$

Одна итерация требует $n^2 + 2n$ операций умножения и $n^2 + 2n$ операций сложения. Для реализации решения необходима память в $n^2 + 2n$ слов. В случае, когда матрица, для которой разыскиваются собственные значения, является порождающей (см. п. 1 параграфа II.2), вместо матрицы можно задавать процедуру умножения матрицы на вектор, и в этом случае степенной метод требует хранить в памяти ЭВМ $2n$ чисел. Естественно, что эта экономия памяти достигается за счет некоторого увеличения времени решения задачи на вычислительной машине.

Ранее для упрощения доказательства нами предполагалось, что степенной метод применим для матриц простой структуры, имеющих изолированное собственное значение. Однако при решении практических задач от этого ограничения можно отказаться. В работе [103] рассмотрены и теоретически обоснованы варианты степенного метода и в случаях, когда несколько линейно независимых собственных векторов будут соответствовать одному собственному значению, когда максимальные собственные значения составляют комплексно-сопряженную пару корней, когда модули двух действительных собственных значений λ_n и λ_{n-1} равны или близки между собой и т. д.

Априори по внешнему виду матрицы нельзя определить такие ее свойства, которые позволили бы выбрать тот или иной вариант степенного метода. Поэтому при необходимости в определении максимального собственного значения и соответствующего ему собственного вектора степенным методом реализуют вычисления по формулам (III.46),

(III.48) и по поведению итерационного процесса судят о расположении и о некоторых свойствах собственных значений, позволяющих выбрать тот или иной вариант степенного метода.

Если итерационный процесс сходится слишком медленно, то это может свидетельствовать о наличии близких по модулю корней. Если итерации колеблются и меняют знак, то это свидетельствует о наличии комплексной пары собственных значений. И в том, и в другом случаях можно использовать модификации степенного метода, рассмотренные в работе [103]. В любом случае полезно продублировать вычисления, начав итерации с нового начального приближения.

Степенной метод сходится очень медленно. Обычно его используют с различными приемами ускорения сходимости [97]. В случае симметричной матрицы на основе степенного метода можно разработать такой метод, который обладает более высокой скоростью сходимости собственных значений, чем степенной. Этот метод получил название метода скалярных произведений.

3. Метод скалярных произведений целесообразно использовать при нахождении максимального по модулю собственного значения и принадлежащего ему собственного вектора симметричной матрицы.

Вычислительная схема может быть реализована следующим образом:

$$y^{(k)} = A \tilde{y}^{(k-1)}, \quad \tilde{y}^{(k)} = \frac{1}{\sqrt{(y^{(k)}, y^{(k)})}} y^{(k)}, \quad \mu_k = \frac{(y^{(k)}, y^{(k)})}{(y^{(k-1)}, y^{(k)})}. \quad (\text{III.52})$$

Условием окончания итерационного процесса может служить, например,

$$|\mu_k - \mu_{k-1}| \leq \varepsilon |\mu_k|.$$

Пример 7. Пусть требуется найти максимальное собственное значение матрицы (III.48) методом скалярных произведений.

Решение. Выбрав в качестве начального приближения $y^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, вычислим

$$y^{(1)} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \\ 4 \end{bmatrix}, \quad \tilde{y} = \frac{1}{9} \begin{bmatrix} 4 \\ 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 0,4444 \\ 0,7778 \\ 0,4444 \end{bmatrix},$$

$$\mu_1 = \frac{16 + 49 + 16}{4 + 7 + 4} = \frac{81}{15} = 5,4.$$

Результаты приближений, полученных в ходе итерационного процесса, приведены в табл. 5.

Из таблицы видно, что для получения максимального собственного значения с погрешностью 10^{-3} достаточно трех итераций.

Покажем сходимость метода скалярных произведений для сим-

Т а б л и ц а 5

μ, y_i	k			
	0	1	2	3
μ	—	5,400 0	5,372 4	5,372 3
y_1	1	0,444 4	0,455 1	0,454 4
y_2	1	0,777 8	0,765 4	0,766 2
y_3	1	0,444 4	0,455 1	0,454 3

метричных матриц. Так как симметричная матрица обладает полной ортогональной системой собственных векторов $\{v_j\}$, $j = 1, 2, \dots, n$, которые могут быть приняты в качестве базиса в n -мерном евклидовом пространстве, то произвольный вектор $y^{(0)}$ может быть представлен в виде

$$y^{(0)} = c_1 v_1 + c_2 v_2 + \dots + c_n v_n.$$

Учитывая, что $(v_i, v_j) = \delta_{ij}$ (δ_{ij} — символ Кронекера), имеем

$$(y^{(k)}, y^{(k)}) = (A^k y^{(0)}, A^k y^{(0)}) = (A^{2k} y^{(0)}, y^{(0)}) = (c_1 \lambda_1^{2k} v_1 + \dots + c_n \lambda_n^{2k} v_n, c_1 v_1 + c_2 v_2 + \dots + c_n v_n) = c_1^2 \lambda_1^{2k} + c_2^2 \lambda_2^{2k} + \dots + c_n^2 \lambda_n^{2k}.$$

Аналогично,

$$(y^{(k-1)}, y^{(k)}) = c_1^2 \lambda_1^{2k-1} + c_2^2 \lambda_2^{2k-1} + \dots + c_n^2 \lambda_n^{2k-1}.$$

Далее

$$\mu_k = \frac{(y^{(k)}, y^{(k)})}{(y^{(k-1)}, y^{(k)})} = \frac{c_1^2 \lambda_1^{2k} + c_2^2 \lambda_2^{2k} + \dots + c_n^2 \lambda_n^{2k}}{c_1^2 \lambda_1^{2k-1} + c_2^2 \lambda_2^{2k-1} + \dots + c_n^2 \lambda_n^{2k-1}} = \lambda_n + O\left(\frac{\lambda_{n-1}}{\lambda_n}\right)^{2k}. \quad (\text{III.53})$$

Отсюда видно, что

$$\lim_{k \rightarrow \infty} \mu_k = \lambda_n.$$

Одна итерация метода требует $n^2 + 3n$ операций умножения, $n^2 + 2n$ операций сложения. Для его реализации необходимо хранить в памяти ЭВМ количество чисел $n^2 + 2n$. В случае порождающихся матриц можно задавать процедуру умножения матрицы на вектор, тогда требуется хранить в памяти количество чисел $2n$.

Из (III.53) видно, что в методе скалярных произведений число итераций, нужных для определения λ_n с данной точностью, почти вдвое меньше, чем в степенном методе. Сходимость по собственным векторам такая же, как и в степенном методе.

4. Метод обратных итераций. В некоторых практических задачах возникает необходимость в нахождении минимального по модулю собственного значения и принадлежащего ему собственного вектора. Для решения задачи можно использовать модификацию степенного метода, итерируя вектор обратной матрицей

$$y^{(k)} = A^{-1} y^{(k-1)}. \quad (\text{III.54})$$

Таким образом, можно вычислить максимальное собственное значение матрицы A^{-1} . Его обратное значение равно минимальному собственному значению матрицы A . При реализации этого метода на практике нет необходимости обращаться матрицу A , так как это требует большого числа арифметических операций и портит исходную структуру матрицы (например, если исходная матрица ленточная, то обратная матрица будет с полным заполнением), поэтому вместо итераций по формуле (III.54) на каждом шаге итерационного процесса можно решать систе-

му линейных алгебраических уравнений

$$A y^{(k)} = y^{(k-1)}.$$

При этом, как и в степенном методе, на каждом шаге необходима нормировка вектора.

Метод обратных итераций, как и степенной, обладает линейной сходимостью, поэтому на практике используют те или иные приемы, ускоряющие сходимость его.

Рассмотрим один из возможных вариантов ускорения сходимости метода обратных итераций, известный под названием метода Виландта. В этом методе по исходной матрице A строится некоторая вспомогательная матрица $B = (A - \mu_i E)^{-1}$ с собственными числами $v_k = 1/(\lambda_k - \mu_i)$, где μ_i — приближение к i -му собственному значению λ_i матрицы A . Затем степенным методом вычисляют максимальное собственное значение матрицы B . Так как при хорошем начальном приближении μ_i значение v_i преобладает над остальными собственными значениями, степенной метод сходится за несколько итераций. После получения собственного значения v_i^k (k — номер итерации) матрицы B собственное значение λ_i матрицы A вычисляют по формуле

$$\lambda_i = \mu_i + \frac{1}{v_i^k},$$

а $y^{(k)}$ является собственным вектором, соответствующим этому собственному значению. При этом, как и в методе обратных итераций, нет необходимости предварительно обращать матрицу $(A - \mu_i E)$, а достаточно на каждом шаге степенного метода решать систему линейных алгебраических уравнений

$$(A - \mu_i E) y^{(k)} = y^{(k-1)}.$$

Вычислительная схема метода обратных итераций для уточнения i -го собственного значения матрицы A и вычисления соответствующего ему собственного вектора может быть реализована следующим образом.

До начала итерационного процесса задают приближение μ к λ_i , начальное приближение $y^{(0)}$ к собственному вектору y_i , относительную погрешность получаемого собственного значения ε . Тогда для $k = 1, 2, \dots$

$$(A - \mu E) \tilde{y}^{(k)} = y^{(k-1)}, \quad v^{(k)} = \max_{1 \leq i \leq n} \{\tilde{y}_i^{(k)}\}, \quad y^{(k)} = \frac{\tilde{y}^{(k)}}{v^{(k)}}.$$

Если окажется, что все значения $y_i^{(k)} < 0$, то в качестве $v^{(k)}$ следует брать минимальную компоненту $\tilde{y}_i^{(k)}$. Если $|v^{(k)} - v^{(k-1)}| < \varepsilon |v^{(k)}|$, то полагаем

$$\lambda_i = \mu + \frac{1}{v^{(k)}}, \quad \text{а } y_i = y^{(k)}.$$

Если неравенство не выполнено, то итерационный процесс продолжают до его выполнения.

Рассмотренный нами метод Виландта может использоваться в различных вариантах. Им можно уточнять любые действительные собственные значения матрицы или собственные значения и соответствующие им собственные векторы. Этот метод позволяет по известному собственному значению найти принадлежащий ему собственный вектор. Методом Виландта можно вычислить минимальное собственное значение и соответствующий ему вектор матрицы A . Для этого предварительно проводят несколько обратных итераций без сдвига. Получив таким образом начальное приближение к минимальному собственному значению, уточняют его методом Виландта.

Преимущества метода являются достаточно быстрая сходимость, возможность сохранения исходной структуры матрицы, простота реализации на ЭВМ. К недостаткам его следует отнести необходимость на каждом шаге процесса решать систему линейных алгебраических уравнений. Впрочем, если известно треугольное разложение матрицы, то на каждом шаге необходимо выполнять лишь обратный ход.

Пример 8. Пусть для заданной матрицы

$$\begin{bmatrix} 33 & 16 & 72 \\ -24 & -10 & -57 \\ -8 & -4 & -17 \end{bmatrix}$$

известно приближение ко второму собственному значению $\mu_2 = 2,1$. Требуется уточнить второе собственное значение и принадлежащий ему собственный вектор.

Решение. Построим матрицу

$$(A - \mu_2 E) = \begin{bmatrix} 30,9 & 16 & 72 \\ -24 & -12,1 & -57 \\ -8 & -4 & -19,1 \end{bmatrix}.$$

Выбрав начальное приближение ко второму собственному вектору $y^0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ и решив систему линейных алгебраических уравнений

$$(A - \mu_2 E) \tilde{y}^{(1)} = y^{(0)},$$

получим $\tilde{y}^{(1)} = \begin{bmatrix} -202,93 \\ 168,79 \\ 49,596 \end{bmatrix}$. Нормируем $\tilde{y}^{(1)}$, разделив все его компоненты на макси-

мальную по модулю компоненту $v^{(1)} = -202,93$, имеем $y^{(1)} = \begin{bmatrix} 1 \\ -0,83176 \\ -0,24220 \end{bmatrix}$. Далее

решаем систему уравнений $(A - \mu_2 E) \tilde{y}^{(2)} = y^{(1)}$ и это решение нормируем. В результате получаем $v^{(2)} = -15,731$, $y^{(2)} = \begin{bmatrix} 1 \\ -0,81168 \\ -0,24968 \end{bmatrix}$ и т. д. И наконец, имеем

$v^{(5)} = -9,9991$, а $v^{(6)} = -10,000$, $y^{(6)} = \begin{bmatrix} 1 \\ -0,8125 \\ -0,25 \end{bmatrix}$. Таким образом, с точностью

до трех десятичных знаков значение $v^{(6)}$ является максимальным собственным зна-

чением матрицы $(A - \mu_2 E)^{-1}$. Вторым собственным значением матрицы A является

$$\lambda_2 = \mu_2 + \frac{1}{v^{(6)}} = 2,1 - 0,1 = 2,$$

а $y^{(6)}$ — приближенным собственным вектором, соответствующим λ_2 .

Отметим, что точными собственными значениями матрицы являются $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 3$.

5. Метод Ланцоша [164] до недавнего времени был известен как алгоритм приведения симметричной матрицы A к трехдиагональной матрице T . Основой алгоритма служит трехчленное рекуррентное соотношение

$$q_{j+1}\beta_j = Aq_j - \alpha_j q_j - q_{j-1}\beta_{j-1}, \quad j = 1, \dots, n, \quad q_0 = 0, \quad (\text{III.55})$$

которое получается из следующего матричного равенства:

$$Q_n T_n = A Q_n,$$

где $Q_n = [q_1, q_2, \dots, q_n]$ — ортогональная матрица, T_n — трехдиагональная матрица

$$T_n = \begin{vmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \dots & 0 & 0 \\ 0 & \beta_2 & \alpha_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} & 0 \\ 0 & 0 & 0 & \dots & \beta_{n-1} & \alpha_n \end{vmatrix}.$$

Если T_n — неразложимая матрица, то, не ограничивая общности рассуждений, можно считать $\beta_j > 0$ (трехдиагональная матрица называется неразложимой, если все $\beta_j \neq 0$). Тогда соотношение (III.55) позволяет записать следующий алгоритм построения матрицы T_n по матрице A .

Задается начальный вектор $\|q_1\| = 1$, затем для $j = 1, 2, \dots, n$ повторяется следующее:

1. $u_j = Aq_j$,
2. $\alpha_j = (q_j, Aq_j)$,
3. $r_j = u_j - \alpha_j q_j - q_{j-1}\beta_{j-1}$, $q_0 = 0$,
4. $\beta_j = \|r_j\|$,
5. $q_{j+1} = r_j/\beta_j$.

После n шагов в точной арифметике получим трехдиагональную матрицу T_n , ортогонально подобную исходной матрице A , и, следовательно, задача определения собственных значений матрицы A свелась к задаче нахождения собственных значений для трехдиагональной матрицы.

Срыв процесса может произойти, если $\beta_j = 0$. Это произойдет в том случае, если вектор q_1 ортогонален по крайней мере одному собственному вектору A . Если $\beta_j = 0$, то в качестве q_{j+1} можно взять любой единичный вектор, ортогональный к предыдущим векторам q , и продолжать процесс.

Пример 9. Привести по методу Ланцоша матрицу

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

к трехдиагональному виду.

Решение. Возьмем в качестве начального вектор $q_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$.

Далее, согласно (III.56) получаем для $j = 1$

$$u_1 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}, \quad \alpha_1 = 1, \quad r_1 = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}, \quad \beta_1 = \sqrt{5}, \quad q_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix},$$

для $j = 2$

$$u_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} 5 \\ 8 \\ 5 \end{bmatrix}, \quad \alpha_2 = \frac{21}{5}, \quad r_2 = \frac{2}{5\sqrt{5}} \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix}, \quad \beta_2 = \frac{2}{5}, \quad q_3 = \frac{1}{\sqrt{5}} \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix},$$

для $j = 3$

$$u_3 = \frac{1}{\sqrt{5}} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \alpha_3 = -\frac{1}{5}.$$

Таким образом,

$$T_3 = \begin{bmatrix} 1 & \sqrt{5} & 0 \\ \sqrt{5} & \frac{21}{5} & \frac{2}{5} \\ 0 & \frac{2}{5} & -\frac{1}{5} \end{bmatrix}.$$

Однако при реализации алгоритма (III.56) на практике для больших A строгая ортогональность векторов q_i обычно утрачивается. Основной причиной этого является конечная разрядность ЭВМ, что в свою очередь приводит к ошибкам округления. Одним из возможных выходов в этом случае является использование метода Ланцоша с очень точной переортогонализацией. Но такая процедура дорогостояща.

В последнее время благодаря теории Каниэля — Саада [73], метод Ланцоша стал известен как алгоритм определения нескольких крайних собственных значений больших матриц. Речь идет об усеченном варианте алгоритма Ланцоша, в котором собственные значения усеченной трехдиагональной матрицы T_j ($j < n$) будут приближать искомые собственные числа исходной матрицы. Основным здесь является вопрос: когда закончить итерации Ланцоша (на каком j остановиться)? Ответ на него можно дать, трактуя алгоритм Ланцоша как естественный способ реализации процедуры Релея — Ритца для последовательности подпространств Крылова, благодаря которой появ-

ляется возможность контролировать на каждом шаге точность искомых собственных значений. Для реализации этой процедуры продолжим процесс (III.56).

6. Для каждого $j = 1, 2, \dots$ решаем задачу на собственные значения:

$$T_j z' = \theta' z'.$$

7. Вычисляем

$$y^{(j)} = Q_j z^{(j)}, \quad \text{где } Q_j = (q_1, q_2, \dots, q_j).$$

Вычисляются только те θ_i, z_i, y_i , которые нужны.

Пара $(\theta_i^{(j)}, y_i^{(j)})$ называется аппроксимирующей парой Релея — Ритца для собственной пары (λ_i, x_i) исходной матрицы A .

8. Если точность искомых собственных значений удовлетворительна, процесс заканчиваем. Точность контролируется с помощью следующей оценки:

$$|\theta - \lambda| \leq \|Ay - \theta y\| / \|y\|.$$

Если в процессе вычислений получим $\beta_j = 0$, это означает, что $AQ_j = Q_j T_j$, и вычисление заканчиваем. При отыскании нескольких собственных значений такой ранний срыв процесса является желаемым результатом, поскольку каждое собственное значение T_j будет собственным значением A .

При реализации алгоритма Ланцоша в оперативной памяти необходимо хранить только три n -мерных вектора. Матрица может быть представлена подпрограммой, которая по заданному значению x вычисляет вектор Ax . Стоимость одного шага алгоритма в основном определяется формированием вектора Aq . Фиксировать заранее номер последнего шага ($j = m$) нет необходимости. Процесс продолжается до тех пор, пока нужные пары Ритца (y_i, λ_i) , $i = 1, \dots, p$, не станут удовлетворительными по точности. Типичными значениями, которые можно иметь в виду, являются $p = 10$, $m = 300$, $n = 10^4$.

В настоящее время существуют и продолжают появляться различные модификации алгоритма Ланцоша. Обзор их можно найти в работах [34, 73].

6. Метод одновремених итераций. Метод одновремених итераций [186] является обобщением степенного метода для случая определения нескольких крайних собственных значений произвольных матриц. Опишем наиболее простую схему, реализующую метод одновремених итераций для симметричных матриц.

Перед началом вычислений задаем матрицу $Q^{(0)}$ с размерами $n \times p$, состоящую из p ортонормированных векторов-столбцов длины n . Затем для $k = 1, 2, \dots$ выполняем следующее:

1) вычисляем

$$\tilde{Q}^{(k)} = A Q^{(k-1)}.$$

2) ортогонализируем столбцы матрицы $\tilde{Q}^{(k)}$, используя для этого, например, ортогонально-треугольное разложение

$$\tilde{Q}^{(k)} = Q^{(k)} U^{(k)}, \quad (\text{III.57})$$

где $Q^{(k)}$ — ортогональная, $U^{(k)}$ — верхняя треугольная матрицы.

Предполагая, что начальные вектор-столбцы матрицы $Q^{(0)}$ не ортогональны к искомым собственным векторам, при $k \rightarrow \infty$ вектор-столбцы матрицы $Q^{(k)}$ будут стремиться к соответствующим собственным векторам, а $U^{(k)}$ — к диагональной матрице порядка p , диагональные элементы которой являются собственными значениями матрицы A .

Пример 9. Найти два максимальных собственных значения матрицы

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

используя одновременную итерацию двух векторов.

Решение. В качестве начального приближения берем матрицу $Q^{(0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

Далее вычисляем

$$\begin{aligned} \tilde{Q}^{(1)} &= \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 1 & 1 \end{bmatrix}, \quad Q^{(1)} = \begin{bmatrix} 1/\sqrt{6} & 1/\sqrt{3} \\ 2/\sqrt{6} & -1/\sqrt{3} \\ 1/\sqrt{6} & 1/\sqrt{3} \end{bmatrix}, \quad U^{(1)} = \begin{bmatrix} \sqrt{6} & 10/\sqrt{6} \\ 0 & \sqrt{3}/3 \end{bmatrix}; \\ \tilde{Q}^{(2)} &= \begin{bmatrix} 6/\sqrt{6} & 0 \\ 10/\sqrt{6} & 1/\sqrt{3} \\ 6/\sqrt{6} & 0 \end{bmatrix}, \quad Q^{(2)} = \begin{bmatrix} 3/\sqrt{43} & 5/\sqrt{86} \\ 5/\sqrt{43} & -6/\sqrt{86} \\ 3/\sqrt{43} & 5/\sqrt{86} \end{bmatrix}, \quad U^{(2)} = \begin{bmatrix} \sqrt{86}/3 & 5/\sqrt{142} \\ 0 & \sqrt{6}/43 \end{bmatrix}; \\ \tilde{Q}^{(3)} &= \begin{bmatrix} 16/\sqrt{43} & -2/\sqrt{86} \\ 27/\sqrt{43} & 2/\sqrt{86} \\ 3/\sqrt{43} & -2/\sqrt{86} \end{bmatrix}, \quad Q^{(3)} = \begin{bmatrix} 16/\sqrt{1241} & -1161/\sqrt{5391684} \\ 27/\sqrt{1241} & 1376/\sqrt{5391684} \\ 16/\sqrt{1241} & -1161/\sqrt{5391684} \end{bmatrix}, \\ U^{(3)} &= \begin{bmatrix} \sqrt{1241/43} & 10/\sqrt{106726} \\ 0 & \sqrt{7396/53363} \end{bmatrix}. \end{aligned}$$

После трех итераций на диагонали матрицы $U^{(3)}$ получили приближения $\lambda_3^{(3)} = 5,372$, $\lambda_2^{(3)} = 0,37$ к двум максимальным собственным значениям матрицы A с точностью до двух значащих цифр. Метод обладает всеми достоинствами и недостатками, что и степенной метод. Скорость сходимости метода одновременных итераций, описанного выше, как и степенного метода, невысокая и определяется величиной λ_{i-1}/λ_i . Эту ситуацию позволяет исправить сочетание одновременных итераций с процедурой Релея — Ритца [73].

В этом случае вычислительная схема метода будет следующей.

1. Вычисляем

$$\tilde{Q}^{(k)} = A Q^{(k-1)}.$$

2. Ортогонализируем столбцы

$$\tilde{Q}^{(k)} \rightarrow S^{(k)}$$

(для этого можно использовать, например, модифицированный процесс Грамма — Шмидта).

3. Формируем матрицу

$$H^{(k)} = (S^{(k)})^T A S^{(k)}.$$

4. Решаем задачу на собственные значения

$$H^{(k)} V^{(k)} = V^{(k)} \Lambda^{(k)}. \quad (\text{III.58})$$

5. Вычисляем приближение Ритца к собственным векторам

$$Q^{(k)} = S^{(k)} V^{(k)}. \quad (\text{III.59})$$

Если вектор-столбцы $Q^{(0)}$ не ортогональны к искомым собственным векторам, то при $k \rightarrow \infty$ столбцы $q_i^{(k)}$ матрицы $Q^{(k)}$ приближают собственные векторы матрицы A , а элементы $\lambda_i^{(k)}$, $i = 1, \dots, p$, диагональной матрицы $\Lambda^{(k)}$ — соответствующие собственные значения.

Итерационный процесс заканчиваем, как только элементы матрицы $\Lambda^{(k)}$ стабилизируются в процессе счета. При этом величину абсолютной ошибки для p собственных чисел можно оценить через p векторов невязки $r_i^{(k)} = A q_i^{(k)} - \lambda_i^{(k)} q_i^{(k)}$, $i = 1, 2, \dots, p$. Для этого необходимо вычислить $\|r_i^{(k)}\|$. Каждый интервал $(\lambda_i^{(k)} - \|r_i^{(k)}\|, \lambda_i^{(k)} + \|r_i^{(k)}\|)$ содержит собственное число матрицы A . Если некоторые интервалы перекрываются, то требуется несколько больше работы, чтобы гарантировать аппроксимацию p собственных чисел (см., например, [73]).

Следует заметить, что для вычисления p наибольших по модулю собственных значений $(\lambda_n, \dots, \lambda_{n-p+1})$ матрицы A для ускорения сходимости необходимо одновременно проводить итерации с l ($l > p$) векторами, и их число существенно влияет на свойства сходимости.

(Например, в [4] рекомендуется l брать равным $\min(2p, p + 8)$.) Действительно, скорость сходимости итерационного процесса ограничена величинами $\left| \frac{\lambda_{n-l+1}}{\lambda_{n-p+1}} \right|$ и $\exp \left(-\operatorname{arsh} \left(\left| \frac{\lambda_{n-p+1}}{\lambda_{n-l+1}} \right| \right) \right)$.

Причем она близка к первой величине, если отношение $\frac{\lambda_n}{\lambda_{n-p+1}}$ велико, и близка ко второй, если указанное отношение порядка единицы. Следовательно, этот метод будет сравнительно быстросходящимся для тех матриц, наибольшие собственные значения которых близки друг к другу.

III.5. НЕКОТОРЫЕ ПОДХОДЫ К РЕШЕНИЮ ОБОБЩЕННОЙ ЗАДАЧИ НА СОБСТВЕННЫЕ ЗНАЧЕНИЯ

1. Сведение к обычной задаче на собственные значения. Как отмечалось в п. 1 параграфа III.1, некоторые прикладные задачи сводятся к нахождению таких значений λ , при которых существуют отличные от нулевого решения системы линейных алгебраических уравнений

$$A v = \lambda B v, \quad |B| \neq 0. \quad (\text{III.60})$$

Эти значения параметра λ называются собственными значениями или собственными числами матрицы A относительно матрицы B , а

соответствующие этим λ отличные от нулевого решения $\mathbf{v}$ системы (III.60) называются собственными векторами матрицы A относительно матрицы B .

Так как $|B| \neq 0$, то существует B^{-1} и задача (III.60) может быть сведена к задаче

$$B^{-1}A\mathbf{v} = \lambda\mathbf{v}. \quad (\text{III.61})$$

Для решения задачи (III.61) могут быть применены методы решения полной или частичной проблемы собственных значений, рассмотренные выше. Но такой прием не всегда целесообразен: много арифметических операций требует обращение матрицы B ; если в задаче (III.60) A и B были симметричными матрицами, то матрица $B^{-1}A$ может оказаться несимметричной, и т. д. Поэтому если в (III.60) A и B симметричные матрицы и кроме того B положительно определенная, то вместо (III.60) можно рассматривать задачу

$$B^{-1/2}AB^{-1/2}\mathbf{z} = \lambda\mathbf{z}, \quad (\text{III.62})$$

где $\mathbf{z} = B^{1/2}\mathbf{v}$.

Матрица $B^{-1/2}AB^{-1/2}$ симметрична. Действительно, пусть λ — собственное значение A относительно B , $\mathbf{v}$ — принадлежащий ему собственный вектор. Так как B положительно определенная матрица, то существуют положительно определенные $B^{1/2}$ и $B^{-1/2}$, и из (III.60) можно записать

$$B^{-1/2}A\mathbf{v} = \lambda B^{1/2}\mathbf{v}.$$

Последнее равенство можно переписать так:

$$B^{-1/2}AB^{-1/2}(B^{1/2}\mathbf{v}) = \lambda B^{1/2}\mathbf{v}$$

или в виде (III.62), где λ — собственное значение, а $\mathbf{z} = B^{1/2}\mathbf{v}$ — соответствующий ему собственный вектор задачи (III.62).

Используя рассмотренные свойства, можно строить методы решения различных обобщенных задач на собственные значения.

Если в задаче (III.60) матрица B симметрична и положительно определенная, то, используя известные алгоритмы решения полной проблемы собственных значений для симметричных матриц (например, QL -метод в сочетании с методом Хаусхолдера), эту задачу можно свести к виду

$$\tilde{A}\mathbf{y} = \lambda D\mathbf{y}, \quad (\text{III.63})$$

где $\tilde{A}$ — матрица, полученная из A теми же преобразованиями, которые применялись к матрице B , D — диагональная матрица. Одновременно такие же преобразования проводятся над строками некоторой вспомогательной матрицы P , которая первоначально является единичной. Таким образом, получаем

$$\mathbf{y} = P^T\mathbf{v} \quad (\text{III.64})$$

Так как для диагональной матрицы легко вычислить

$$D^{-1/2} = \begin{vmatrix} \frac{1}{\sqrt{d_1}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{d_2}} & \dots & 0 \\ 0 & 0 & \dots & \frac{1}{\sqrt{d_n}} \end{vmatrix},$$

то задачу (III.63), (III.64) легко привести к виду (III.62):

$$D^{-1/2}\tilde{A}D^{-1/2}\mathbf{z} = \lambda\mathbf{z}, \quad \mathbf{z} = D^{1/2}\mathbf{y}. \quad (\text{III.65})$$

Если $D^{-1/2}\tilde{A}D^{-1/2}$ — симметричная матрица, то для решения задачи (III.65) можно использовать рассмотренные выше метод Якоби или QL -метод в сочетании с методом Хаусхолдера или Гивенса. Если матрица A — несимметрична, то матрица $D^{-1/2}\tilde{A}D^{-1/2}$ также будет несимметричной, и для решения задачи (III.65) можно использовать методы нахождения собственных значений несимметричных матриц, рассмотренные в работе [97].

Другой способ приведения обобщенной задачи $A\mathbf{x} = \lambda B\mathbf{x}$ к обычной заключается в следующем.

Вычисляем LL^T разложение для симметричной положительно определенной матрицы $B = LL^T$. Тогда

$$L^{-1}AL^{-T}(L^T\mathbf{x}) = \lambda L^T\mathbf{x}$$

и обобщенная задача сводится к обычной задаче

$$\tilde{A}\mathbf{y} = \lambda\mathbf{y}, \quad \tilde{A} = L^{-1}AL^{-T}.$$

Собственный вектор $\mathbf{x}$ восстанавливается по формуле $\mathbf{x} = L^{-T}\mathbf{y}$. Если матрица A была симметричной, то симметричной будет и матрица $\tilde{A}$.

2. Приведение к обобщенной форме Шура. Основу преобразования пары матриц A, B к обобщенной форме Шура составляет утверждение, следующее из [172]: существуют ортогональные матрицы Q, Z такие, что матрица $H = QAZ$ — квазитреугольная, $R = QBZ$ — треугольная (пару матриц H, R , одна из которых квазитреугольная, а другая треугольная, называют обобщенной формой Шура).

Таким образом, обобщенную задачу на собственные значения

$$A\mathbf{x} = \lambda B\mathbf{x} \quad (\text{III.66})$$

можно свести к эквивалентной задаче

$$H\mathbf{y} = \lambda R\mathbf{y}, \quad (\text{III.67})$$

собственные значения которой совпадают с собственными значениями (III.66), а собственные векторы связаны соотношением $\mathbf{x} = Z\mathbf{y}$.

Для численной реализации этого преобразования обычно используется QZ -алгоритм [172], который является обобщением QR -алгоритма для пары матриц. Практически QZ -алгоритм применяется к паре

матриц, предварительно преобразованных к обобщенной форме Хессенберга (пару матриц называют обобщенной формой Хессенберга, если одна из них — матрица Хессенберга, а другая — треугольная матрица).

Таким образом, при решении обобщенной задачи на собственные значения можно реализовать следующий алгоритм.

1. Ортогонально эквивалентными преобразованиями $H_1 = Q_1 A Z_1$, $R_1 = Q_1 B Z_1$ задача (III.66) сводится к виду

$$H_1 y_1 = \lambda R_1 y_1, \quad (\text{III.68})$$

где H_1 — матрица Хессенберга, R_1 — треугольная матрица того же вида, $y_1 = Z_1^{-1} x$. Это преобразование можно построить посредством конечного вычислительного процесса, используя ортогональные преобразования отражения и вращения.

2. Используя QZ-алгоритм, задачу (III.68) преобразуем к виду

$$H y = \lambda R y. \quad (\text{III.69})$$

Здесь $H = Q H_1 Z$ — квазиреугольная матрица, $R = Q R_1 Z$ — треугольная, вектор y_1 вычисляется по формуле $y_1 = Z y$. Фактически QZ-алгоритм эквивалентен QR-алгоритму для матрицы $H_1 R_1^{-1}$ без явного формирования последней.

3. Решение задачи (III.69) очевидно состоит в решении обобщенной проблемы собственных значений для матрицы первого либо второго порядка. Собственные векторы задачи (III.66) восстанавливаются по формуле $x = Z_1 Z y$. Поскольку выполняются исключительно ортогональные преобразования, на точность вычислений не очень влияет такое неприятное обстоятельство, как вырожденность B .

Для определения по описанному алгоритму всех собственных значений требуется около $20n^3$ арифметических операций. Временные затраты менее обременительны, чем необходимость использования двух дополнительных массивов с размерами $n \times n$. Однако для малых значений n оба вида затрат не будут очень серьезными. При реализации этого алгоритма симметрия матрицы теряется, поэтому для симметричных матриц экономичнее использовать алгоритмы, учитывающие этот факт.

Методы нахождения собственных значений и принадлежащих им собственных векторов условно можно разделить на методы, в которых вначале строится характеристический полином (см. п. 1 параграфа III.1), затем находятся корни нелинейного алгебраического уравнения n -й степени; и методы, в которых собственные значения и принадлежащие им собственные векторы вычисляются без использования характеристического полинома.

Наиболее распространены в настоящее время методы второго типа. Самым простым методом определения всех собственных значений и собственных векторов плотной симметричной матрицы является метод Якоби, в частности усовершенствованный вариант, предложенный В. В. Воеводиным [15]. Метод отличается компактностью в реализации и высокой точностью вычисляемых собственных значений и векторов.

Преобразование Хаусхолдера в сочетании с QL-методом для плотных симметричных и преобразование Шварца в сочетании с QL-методом для ленточных симметричных матриц — наиболее эффективные алгоритмы определения всех собственных значений и, если необходимо, собственных векторов.

При вычислении нескольких собственных значений (не более, чем $1/2 n$, где n — порядок матрицы) целесообразно использование метода Хаусхолдера для плотных или метода Шварца для ленточных матриц в сочетании с методом бисекций, основанным на свойствах последовательности Штурма для симметричной трехдиагональной матрицы.

Для нахождения максимального собственного значения матрицы и принадлежащего ему собственного вектора целесообразно использовать метод скалярных произведений.

Если нужно найти минимальное собственное значение и соответствующий ему собственный вектор, то для решения этой задачи можно использовать метод обратных итераций. Для уточнения отдельного собственного значения или нахождения по заданному собственному значению соответствующего ему собственного вектора целесообразно использовать метод Виландта.

При определении нескольких собственных значений и соответствующих собственных векторов для больших матриц в некоторых случаях наиболее целесообразно использовать различные модификации метода Ланцоша [73, 176], либо метода одновременных итераций [73, 4, 161].

В ряде случаев для больших разреженных матриц специального вида оказывается целесообразным применение градиентных методов [30, 76, 84, 89, 130], методов покоординатного спуска [162], метода Ньютона [51], метода верхней релаксации [185]. Иногда большие задачи на собственные значения можно с достаточной точностью аппроксимировать малыми [97].

Симметричные матрицы в задаче нахождения собственных значений всегда хорошо обусловлены, и достоверность машинного решения определяется лишь погрешностью в задании исходных данных и методом решения задачи.

Устойчивость решения при нахождении собственных векторов симметричных матриц зависит от близости собственных значений, погрешности в задании исходных данных и метода решения задачи. Учитывая эти факторы, можно дать апостериорную оценку достоверности получаемого машинного решения.

Если требуется определить собственные значения плотной действительной несимметричной матрицы, то следует иметь в виду, что собственные значения могут оказаться весьма чувствительными к малым изменениям элементов матрицы. У таких матриц могут оказаться нелинейные элементарные делители, и в этом случае собственные векторы не порождают полного пространства. Поскольку при вычислениях на ЭВМ возникают ошибки за счет округления чисел, то очень трудно определить, имеет ли матрица нелинейные делители, поэтому всегда при нахождении собственных значений несимметричной матрицы целесообразно искать полное число собственных векторов. Анализируя вычисленные собственные векторы, бывает нетрудно выявить наличие

дефекта матрицы. В этом случае векторы почти параллельны. Следует заметить, что полученные в результате вычисления собственные значения, соответствующие нелинейному элементарному делителю, могут оказаться несовпадающими.

Перед применением любого алгоритма для вычисления собственных значений несимметричных действительных матриц полезно провести масштабирование исходных матрицы (см., например, [98, 174]).

Для определения всех собственных значений плотной действительной матрицы или действительной ленточной матрицы наиболее эффективно сочетание метода Хаусхолдера (для плотных матриц) или метода Шварца (для ленточных матриц), которые преобразуют исходную матрицу к форме Хессенберга с QR -алгоритмом нахождения собственных значений матрицы Хессенберга. При нахождении нескольких наибольших собственных значений можно использовать различные модификации степенного метода [186].

Погрешность в задании исходных данных может толковаться как возмущение исходной матрицы A , т. е. рассмотрение вместо матрицы A некоторой матрицы $A + \Delta A$, где

$$|\Delta A_{ij}| \leq \delta, \quad (i, j = 1, 2, \dots, n).$$

Поэтому полное решение задачи на собственные значения состоит не только в определении собственных значений A , но и в оценке возможных изменений собственных значений всех матриц класса $A + \Delta A$, где элементы матрицы ΔA удовлетворяют рассмотренному выше условию.

После ввода в ЭВМ матрицы $A + \Delta A$ вместо нее будем иметь некоторую другую возмущенную погрешностями машинного перевода матрицу. Применение того или иного метода к этой возмущенной матрице приводит к тому, что вследствие погрешностей округления и погрешности метода получают решение некоторой другой матрицы $A' + F$, где F — матрица возмущения, возникающая в результате погрешности машинной реализации.

Оценкам ошибок, возникающих вследствие неточности задания исходных данных, и ошибкам машинной реализации посвящены многочисленные публикации (см., например, [97, 137, 190] и др.).

Применение интервальной арифметики при решении задач на собственные значения (см., например, [163]) также позволяет оценить достоверность получаемого машинного решения.

Нахождение собственных значений и принадлежащих им собственных векторов обобщенной задачи на собственные значения — более сложная задача, чем обычная.

Если A и B — симметричные матрицы и B — положительно определенная, то сначала обобщенная задача на собственные значения методом Якоби или методом Гивенса (Хаусхолдера) в сочетании с QL -методом сводится к вспомогательной задаче (см. п. 1 параграфа III.5), а затем строится обычная задача на собственные значения (см. п. 2 параграфа III.5), для решения которой применяют рассмотренные выше методы.

Однако существуют методы решения обобщенной проблемы, не свя-

занные со сведением ее к обычной проблеме для другой матрицы. К ним относятся как методы прямого вычисления характеристического полинома $|A - \lambda B| = 0$, так и итерационные методы, в том числе QZ -алгоритм [172].

Если требуется решить частичную обобщенную проблему собственных значений, то для этой цели можно модифицировать степенной метод, метод скалярных произведений, метод обратных итераций, метод Виландта, метод Ланцоша, метод итераций в подпространстве, градиентные методы [30, 73, 130, 161]. Численным методам решения обобщенной проблемы собственных значений посвящена, также, работа [180]. Решение обобщенной проблемы на собственные значения для разреженных матриц и матриц ленточной структуры рассматривалось в работах [73, 179] и др.

В ряде случаев необходимо решать нелинейную задачу на собственные значения, под которой подразумевается нахождение значений параметра λ и соответствующих векторов x таких, что

$$T(\lambda)x = 0.$$

Числа λ называются собственными значениями параметрической матрицы $T(\lambda)$, а векторы x — ее собственными векторами. Здесь $T(\lambda)$ — квадратная матрица n -го порядка, зависящая от числового параметра λ при тех или других аналитических ограничениях. Отметим, что если задача регулярна, то собственные значения являются корнями характеристического уравнения

$$|T(\lambda)| = 0.$$

Важен частный случай полиномиальных матриц

$$T(\lambda) = \lambda^r A_r + \lambda^{r-1} A_{r-1} + \dots + \lambda A_1 + A_0,$$

где A_i — постоянные матрицы n -го порядка.

Численным методам решения некоторых нелинейных задач посвящены работы [156, 53, 145, 184]. Определенный интерес для приложений имеет обратная задача на собственные значения. Суть ее состоит в том, чтобы выбрать значения параметра, от которых зависят элементы матрицы, с тем чтобы спектр выбранной матрицы совпадал с заданным наперед или удовлетворял заранее поставленным иным условиям. Методам решения обратных задач на собственные значения посвящены работы [52, 129 и др.]. Обзор литературы по методам решения различных задач на собственные значения дан в работах [34, 73].

Отметим, что исследования в области задач на собственные значения сосредоточивались вокруг оценки достоверности получаемых на ЭВМ решений, сочетания прямых методов решения, учитывающих специфику матрицы (разреженной, ленточной или профильной структуры и т. д.), с итерационными методами, итерационных методов решения различных задач на собственные значения, создания программных средств нахождения собственных значений и собственных векторов матриц (см., например, [146, 165]).

МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ С ПРЯМОУГОЛЬНЫМИ И КВАДРАТНЫМИ ВЫРОЖДЕННЫМИ МАТРИЦАМИ

IV.1. НЕКОТОРЫЕ ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

1. Обобщенное решение для систем линейных алгебраических уравнений с прямоугольными и квадратными вырожденными матрицами. При описании физических моделей могут возникать системы линейных алгебраических уравнений с прямоугольными и вырожденными квадратными матрицами. Для систем линейных алгебраических уравнений, не обладающих решением с классической точки зрения, вводят понятие обобщенного решения.

Под обобщенным решением (псевдорешением) системы линейных алгебраических уравнений

$$Ax = b, \quad (IV.1)$$

где A — матрица с размерами $m \times n$, b — заданный вектор, x — искомый вектор, понимают вектор u , удовлетворяющий условию

$$\|b - Au\|^2 = \min_x \|b - Ax\|^2 \quad (IV.2)$$

($\|\cdot\|$ — евклидова норма).

Если система (IV.1) имеет классическое решение, то оно совпадает с обобщенным, и при этом $\min_x \|b - Ax\|^2 = 0$. Однако нахождение

векторов, минимизирующих функционал невязки $\|b - Ax\|^2$, имеет смысл и в случае отсутствия классического решения системы (IV.1). Поэтому введение определения обобщенного решения существенно расширяет понятие искомого решения системы (IV.1).

Легко показать, что обобщенные решения и только они удовлетворяют системе линейных алгебраических уравнений

$$A^T A u = A^T b. \quad (IV.3)$$

Если ранг $r(A)$ матрицы A меньше n , то существует бесчисленное множество обобщенных решений, которые образуют подпространство S размерности $n - r$. Наложив то или иное дополнительное условие, из множества обобщенных решений можно выделить единственное обобщенное решение. Так, обобщенное решение x_H с наименьшей евклидовой нормой $\|x_H\| = \min_{u \in S} \|u\|$ называют обобщенным нормальным решением или нормальным псевдорешением. Отметим, что всегда существует нормальное псевдорешение и оно единственно [14].

2. Обращение прямоугольных и вырожденных матриц. Псевдообратная матрица. В п. 6 параграфа I.1 для квадратной и невырожденной матрицы A рассматривалась обратная матрица A^{-1} . Если же матрица

A прямоугольная или квадратная, но вырожденная, то она не имеет классической обратной матрицы. Однако в этом случае может быть введено понятие обобщенной обратной матрицы $A^\#$, которая обладает некоторыми свойствами обратной матрицы и используется при решении некоторых систем линейных алгебраических уравнений. В случае, когда A — квадратная неособенная матрица, обобщенная обратная $A^\#$ совпадает с обратной A^{-1} матрицей.

Обобщенной обратной (псевдообратной) матрицей $A^\#$ [178, 193] для прямоугольной матрицы A с размерами $m \times n$ называют единственную матрицу, удовлетворяющую четырем условиям:

$$\begin{aligned} AA^\#A &= A, \\ A^\#AA^\# &= A^\#, \\ (AA^\#)^* &= AA^\#, \\ (A^\#A)^* &= A^\#A. \end{aligned} \quad (IV.4)$$

Здесь $*$ означает переход к сопряженной матрице. Существование и единственность псевдообратной матрицы показаны в работе [178].

Рассмотрим один из способов построения псевдообратной матрицы. Известно, что любую действительную матрицу A с размерами $m \times n$ можно представить в виде

$$A = U\Sigma V^T, \quad (IV.5)$$

где матрица U сформирована из m ортонормированных собственных векторов матрицы AA^T , матрица V — из n ортонормированных собственных векторов матрицы $A^T A$, матрица Σ с размерами $m \times n$ имеет вид

$$\Sigma = \begin{bmatrix} D & \\ \cdots & \\ 0 & \end{bmatrix}, \quad m \geq n, \text{ или } \Sigma = [D; 0], \quad m < n, \text{ при } D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_t).$$

Диагональные элементы σ являются неотрицательными значениями квадратных корней из общих собственных значений матриц AA^T и $A^T A$ и называются сингулярными числами матрицы A . Допустим, что сингулярные числа упорядочены $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \dots \geq \sigma_t \geq 0$. Если ранг матрицы A равен r , то $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, $\sigma_{r+1} = \dots = \sigma_t = 0$. Разложение вида (IV.5) называют сингулярным разложением матрицы A . Зная сингулярное разложение, можно сразу выписать псевдообратную матрицу

$$A^\# = V\Sigma^\#U^T, \quad (VI.6)$$

$$\text{где } \Sigma^\# = [D^\#; 0], \quad m \geq n, \quad \Sigma^\# = \begin{bmatrix} D^\# & \\ \cdots & \\ 0 & \end{bmatrix}, \quad m < n,$$

$$D^\# = \text{diag}(\sigma_1^\#, \sigma_2^\#, \dots, \sigma_t^\#), \quad \sigma_i^\# = \begin{cases} \frac{1}{\sigma_i}, & \sigma_i > 0 \\ 0, & \text{если } \sigma_i = 0 \end{cases}.$$

Возможны и другие представления разложения (IV.5). Например, при $m \geq n$ можно записать

$$A = U_c D V^T, \quad D = \text{diag}(\sigma_1, \dots, \sigma_n),$$

где матрица U_c сформирована из n ортонормированных собственных векторов, соответствующих наибольшим собственным значениям матрицы AA^T .

Пример 1. Для матрицы

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 3 & 3 \end{bmatrix}$$

найти псевдообратную матрицу $A^\#$.

Решение. Сингулярное разложение матрицы A , как нетрудно убедиться, можно представить в виде

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{6}} & 0 \end{bmatrix} \begin{bmatrix} \sqrt{27} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Таким образом, псевдообратная матрица записывается в виде

$$A^\# = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{27}} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \end{bmatrix} = \\ = \begin{bmatrix} \frac{5}{9} & -\frac{4}{9} & \frac{1}{9} \\ -\frac{4}{9} & \frac{5}{9} & \frac{1}{9} \end{bmatrix}.$$

Зная псевдообратную матрицу $A^\#$, легко вычислить нормальное псевдорешение системы (IV.1), а именно:

$$\mathbf{x} = A^\# \mathbf{b}.$$

Пример 2. Найти нормальное псевдорешение системы линейных алгебраических уравнений

$$2x_1 + x_2 = 4,$$

$$x_1 + 2x_2 = 5,$$

$$3x_1 + 3x_2 = 9.$$

Решение. Так как из примера 1 известна псевдообратная матрица для матрицы заданной системы, нормальное псевдорешение можно получить следующим образом:

$$\mathbf{x} = A^\# \mathbf{b} = \begin{bmatrix} \frac{5}{9} & -\frac{4}{9} & \frac{1}{9} \\ -\frac{4}{9} & \frac{5}{9} & \frac{1}{9} \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

3. Некоторые свойства псевдообратных матриц. Частные случаи псевдообратных матриц [193]. Рассмотрим некоторые свойства псевдообратных матриц и соотношения, связывающие A , $A^\#$ и A^* . Справедливы равенства

$$(A^\#)^\# = A,$$

$$(A^*)^\# = (A^\#)^*,$$

$$(AA^*)^\# = A^* A^\#, \quad (A^* A)^\# = A^\# A^*,$$

$$A^\# = (A^* A)^\# A^* = A^\# (AA^*)^\#,$$

$$(AA^\#)^\# = (AA^\#), \quad (A^\# A)^\# = A^\# A$$

и некоторые другие. Теперь рассмотрим ряд случаев псевдообращения матриц.

Если матрица A невырождена, то псевдообратная матрица $A^\# = A^{-1}$ представляет собой обычную обратную матрицу.

В случае если A — прямоугольная матрица и $m > n$, $r(A) = n$, то

$$A^\# = A_{\text{лев}}^{-1} = (A^* A)^{-1} A^*,$$

т. е. выполнимо левое обращение матрицы и $A^\# A = E$, причем порядок единичной матрицы равен n .

Если матрица A прямоугольная ($n > m$) и $r(A) = m$, то матрица

$$A^\# = A_{\text{прав}}^{-1} = A^* (AA^*)^{-1}$$

есть правая, обратная $AA^\# = E$, причем порядок единичной матрицы m .

Выполнимы равенства

$$(cA)^\# = \frac{1}{c} A^\#, \quad (-A)^\# = -A^\#,$$

где $c \neq 0$ — скаляр.

Если $\mathbf{u}$ вектор-столбец, не равный нулю, то

$$\mathbf{u}^\# = \frac{\mathbf{u}^*}{\mathbf{u}^* \mathbf{u}} = \frac{\mathbf{u}^*}{\|\mathbf{u}\|^2},$$

а для вектора-строки $\mathbf{v}$, не равной нулю,

$$\mathbf{v}^{\#*} = \frac{\mathbf{v}}{\mathbf{v}^* \mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|^2}.$$

Если в матрице A с размерами $m \times n$, $a_{ij} = 1 \forall i, j$, то

$$A^\# = \frac{1}{mn} A^T.$$

Если $A = \begin{bmatrix} B \\ C \end{bmatrix}$ и $BC^* = 0$, то

$$A^\# = [B^\#, C^\#],$$

$$\begin{bmatrix} B \\ 0 \end{bmatrix}^\# = [B^\#, 0].$$

Для $A = [B, C]$ и $B^*C = 0$

$$A^* = \begin{bmatrix} B^* \\ C^* \end{bmatrix},$$

в частности, $[B, 0^*] = \begin{bmatrix} B^* \\ 0 \end{bmatrix}$.

$$\text{Если } A = \begin{bmatrix} B & 0 \\ 0 & C \end{bmatrix}, \text{ то } A^* = \begin{bmatrix} B^* & 0 \\ 0 & C^* \end{bmatrix}.$$

4. Преобразование прямоугольной матрицы A к двухдиагональному виду. Один из численных методов построения псевдообратной матрицы состоит в использовании сингулярного разложения матрицы A . Такое разложение может осуществляться поэтапно. Сначала матрица A численно с помощью ортогональных преобразований Хаусхолдера (см. п. 5 параграфа III.1) может быть приведена к верхней двухдиагональной матрице, которая имеет сингулярные числа, равные сингулярным числам исходной матрицы. Затем, с помощью QR -метода (см. п. 7 параграфа III.3) вычисляются сингулярные числа двухдиагональной матрицы [154].

Рассмотрим, как прямоугольная матрица A с размерами $m \times n$, $m \geq n$, может быть приведена к двухдиагональному виду. Умножая слева и справа исходную матрицу A соответственно на специально подбираемые матрицы отражения $P^{(k)}, Q^{(k)}$, приходят к верхней двухдиагональной форме

$$P^{(n)} \dots P^{(2)} P^{(1)} A Q^{(1)} Q^{(2)} \dots Q^{(n-2)} = \begin{bmatrix} q_1 & e_2 & 0 & \dots & 0 \\ 0 & q_2 & e_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & e_n \\ 0 & 0 & 0 & \dots & q_n \\ \hline & & & & 0 & \dots \end{bmatrix} = J^{(0)}. \quad \text{IV.7}$$

(IV.7)

Процесс преобразования осуществляется по формулам

$$A^{(1)} \equiv A,$$

$$A^{(k+\frac{1}{2})} = P^{(k)} A^{(k)}, \quad k = 1, 2, \dots, n,$$

$$A^{(k+1)} = A^{(k+\frac{1}{2})} Q^{(k)}, \quad k = 1, 2, \dots, n-2.$$

Матрицы отражения $P^{(k)}$ и $Q^{(k)}$ следует выбирать так, чтобы были выполнены условия

$$a_{ik}^{(k+\frac{1}{2})} = 0, \quad i = k+1, \dots, m;$$

$$a_{kj}^{(k+1)} = 0, \quad j = k+2, \dots, n.$$

В этом случае матрицы $P^{(k)}, Q^{(k)}$ будут иметь вид

$$P^{(k)} = E - \frac{u_k u_k^T}{h_k}, \quad h_k = \frac{1}{2} u_k^T u_k,$$

$$u_k^T = \{0, \dots, 0, a_{kk}^{(k)} \pm \sigma_k^{1/2}, a_{k+1,k}^{(k)}, \dots, a_{m,k}^{(k)}\},$$

$$\sigma_k = (a_{kk}^{(k)})^2 + (a_{k+1,k}^{(k)})^2 + \dots + (a_{m,k}^{(k)})^2, \quad k = 1, 2, \dots, n.$$

$$Q^{(k)} = E - \frac{v_k v_k^T}{h_k}, \quad h_k = \frac{1}{2} v_k^T v_k,$$

$$v_k^T = \{0, \dots, 0, a_{k,k+1}^{(k+\frac{1}{2})} \pm \sigma_k^{1/2}, a_{k,k+2}^{(k+\frac{1}{2})}, \dots, a_{k,n}^{(k+\frac{1}{2})}\},$$

$$\sigma_k = (a_{k,k+1}^{(k+\frac{1}{2})})^2 + (a_{k,k+2}^{(k+\frac{1}{2})})^2 + \dots + (a_{k,n}^{(k+\frac{1}{2})})^2, \quad k = 1, \dots, n-2.$$

Знак перед $\sigma_k^{1/2}$ в выражениях для u_k^T и v_k^T следует выбирать таким же, как и знаки $a_{kk}^{(k)}$ и $a_{k,k+1}^{(k+\frac{1}{2})}$ соответственно.

Окончательно, введя обозначения

$$P^T = P^{(n)} P^{(n-1)} \dots P^{(1)}, \quad P = Q^{(1)} Q^{(2)} \dots Q^{(n-2)},$$

можно записать $J^{(0)} = P^T A Q$.

Здесь P и Q — ортогональные матрицы. При таком преобразовании сингулярные числа матрицы $J^{(0)}$ совпадают с сингулярными числами матрицы A .

Пример 3. Привести к двухдиагональному виду квадратную матрицу

$$A = \begin{bmatrix} 4 & 3 & 1 \\ 2 & -2 & 0 \\ 4 & 1 & -1,55 \end{bmatrix}.$$

Решение. На первом этапе вычислений ($k=1$) для $P^{(1)}$ имеем $\sigma_1 = 36$, $u_1^T = [10, 2, 4]$, $h_1 = 60$,

$$P^{(1)} = \frac{1}{15} \begin{bmatrix} -10 & -5 & -10 \\ -5 & 14 & -2 \\ -10 & -2 & 11 \end{bmatrix}, \quad A^{(\frac{3}{2})} = P^{(1)} A^{(1)} = \begin{bmatrix} -6 & -2 & 0,3667 \\ 0 & -3 & -0,1267 \\ 0 & -1 & -1,8033 \end{bmatrix}$$

и для $Q^{(1)}$: $\sigma_1 = 4,1345$, $v_1^T = [0, -4,0334, 0,3667]$, $h_1 = 8,2014$,

$$Q^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0,9836 & 0,1804 \\ 0 & 0,1804 & 0,9836 \end{bmatrix}, \quad A^{(2)} = A^{(\frac{3}{2})} Q^{(1)} = \begin{bmatrix} -6 & 2,0334 & 0,0001 \\ 0 & 2,9279 & -0,6658 \\ 0 & 0,6583 & -1,9541 \end{bmatrix}.$$

На втором этапе вычислений ($k=2$)

$$\sigma_2 = 3,0010, \mathbf{u}_2^T = [0, 5,9289, 0,6583], h_2 = 17,7 \ 926,$$

$$P^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0,9756 & -0,21 \\ 0 & -0,2194 & 0,9756 \end{bmatrix},$$

$$A\left(\frac{5}{2}\right) = P^{(2)}A^{(2)} = \begin{vmatrix} -6 & 2,0334 & 0,0001 \\ 0 & -3,0009 & 1,0783 \\ 0 & 0,0001 & -1,7603 \end{vmatrix}.$$

На третьем этапе имеем для $k = 3$:

$$\sigma_3 = -1,7603, \mathbf{u}_3^T = [0, 0, 2(1,7603)], h_3 = 2(1,7603)^2,$$

$$P^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad A^{(\frac{7}{2})} = P^{(3)} A^{(\frac{5}{2})} = \begin{bmatrix} -6 & 2,0334 & 0,0001 \\ 0 & -3,0009 & 1,0783 \\ 0 & 0,0001 & 1,7603 \end{bmatrix}.$$

Таким образом, с точностью до $1 \cdot 10^{-4}$ нами получена двухдиагональная матрица $A_{\frac{7}{2}}$. Приведем сингулярные числа матрицы A и $A_{\frac{7}{2}}$, вычисленные на ЭВМ «Мир-2» с длиной машинного слова 10 десятичных знаков. Для A имеем: $\sigma_1 = 1,598\ 543\ 907$, $\sigma_2 = 3,084\ 512\ 560$, $\sigma_3 = 6,429\ 069\ 860$, для $A^{(3)}$: $\sigma_1 = 1,598\ 273\ 502$, $\sigma_2 = 3,084\ 471\ 052$, $\sigma_3 = 6,429\ 079\ 511$.

5. Сингулярное разложение двухдиагональной матрицы. Следуя [154], изложим алгоритм сингулярного разложения двухдиагональной матрицы, который вместе с алгоритмом преобразования произвольной матрицы к двухдиагональному виду позже будет использован нами для численного решения систем линейных алгебраических уравнений с прямоуглыми и вырожденными матрицами и для псевдообращения матриц. С помощью QR -метода (см. п. 6 параграфа III.3) можно привести двухдиагональную матрицу $J^{(0)}$ к диагональной форме D , так что выполняется последовательность преобразований

$$J^{(0)} \rightarrow J^{(1)} \rightarrow \dots \rightarrow D, \quad J^{(i-1)} = S^{(i)T} J^{(i)} T^{(i)}, \quad (\text{IV.8})$$

где $S^{(i)}$ и $T^{(i)}$ — ортогональные матрицы, которые выбирают так, чтобы $J^{(i+1)}$ сохраняли двухдиагональную форму, а симметричная трехдиагональная матрица $J^{(i)T} J^{(i)}$ стремилась к диагональному виду.

Для удобства опустим индексы и введем следующие обозначения:

$$J = J^{(i)}, \quad \bar{J} = J^{(i+1)}, \quad S = S^{(i)}, \quad T = T^{(i)}, \quad M = J^T J, \quad \bar{M} = \bar{J}^T \bar{J}.$$

Переход $J \rightarrow \bar{J}$ осуществляется с помощью последовательности преобразований вращения (см. п. 3 параграфа III.3). Таким образом,

$$\bar{J} = \underbrace{S_n^\top S_{n-1}^\top \dots S_2^\top}_{S^\top} J \underbrace{T_2 T_3 \dots T_n}_T. \quad (\text{IV.9})$$

Здесь S_k, T_k — элементарные матрицы вращения вида

$$\begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & \ddots & & & & & & \\ & & & & c_k & -s_k & & & & \\ & & & & s_k & c_k & & & & \\ & & & & & & 1 & & & \\ & & & & & & & \ddots & & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{bmatrix} \begin{matrix} k-1 \\ k \\ k-1 \\ k \end{matrix}$$

причем $s_k^2 + c_k^2 = 1$.

Очевидно, что умножение справа на матрицу вращения изменяет лишь $(k - 1)$ и k столбцы матрицы, а умножение слева на матрицу вращения — лишь $(k - 1)$, k строки.

Коэффициенты c_2, s_2 матрицы T_2 оставим пока неопределенными, в то время как остальные коэффициенты c_k, s_k будем выбирать так, чтобы матрица $\bar{J}$ имела ту же форму, что и J . Следовательно, матрица T_2 не аннулирует ни одного элемента, но добавляет элемент J_{21} , матрица S_2 аннулирует J_{21} , но добавляет J_{13} , матрица T_3 аннулирует J_{13} , но добавляет J_{32} и т. д. и окончательно матрица S_n^T аннулирует $J_{n,n-1}$ и ничего не добавляет.

При этом очевидны следующие соотношения:

$$\bar{J}^T = (S^T J T)^T = T^T J^T S,$$

$$\bar{M} = \bar{J}^T J = T^T J^T S S^T J^T = T^T M^T.$$

Таким образом, ортогональное преобразование (IV.9) эквивалентно преобразованию подобия симметричной трехдиагональной матрицы

$$\overline{M} = T^T M T, \quad (\text{IV.10})$$

где матрица $\bar{M}$ будет также трехдиагональной, поэтому матрицу T_2 , которая пока еще не определена, необходимо выбирать так, чтобы преобразование $M \rightarrow \bar{M}$ было QR -преобразованием со сдвигом, равным s .

Обычно (см. п. 6 параграфа III.3) *QR*-алгоритм со сдвигом можно записать в следующем виде:

$$M - sE = T_s R_s, \quad R_s T_s + sE = \bar{M}_s, \quad (\text{IV.11})$$

где $T_s^* T_s = E$, R_s — верхняя треугольная матрица. Но не обязательно выполнять вычисления по формулам (IV.11). Сдвиг можно осуществить неявным образом. Доказано, например, в работе [97], что определенным выбором T_2 можно добиться того, чтобы преобразование (IV.10) было эквивалентно QR -преобразованию для M с заданным сдвигом s .

Пусть T и T_s ортогональные матрицы такие, что выполняются следующие условия:

$$\{T_s\}_{k,1} = \{T_{k,1}\}, \quad k = 1, 2, \dots, n,$$

т. е. элементы первого столбца T_s равны элементам первого столбца T и

$$\bar{M}_s = T_s^T M T_s, \quad \bar{M} = T^T M T. \quad (IV.12)$$

Тогда если поддиагональные элементы матрицы M ненулевые, то матрица $\bar{M}$ связана с $\bar{M}_s$ следующим образом:

$$\bar{M} = D \bar{M}_s D, \quad (IV.13)$$

где D — диагональная матрица, элементы которой равны ± 1 .

Следовательно, задача заключается лишь в том, чтобы первый столбец матрицы T выбрать равным первому столбцу матрицы T_s . Далее, имеем

$$T_s = (M - sE) R_s^{-1}.$$

Учитывая тот факт, что обратная матрица к верхней треугольной будет также верхней треугольной, можно сделать вывод, что первый столбец искомой матрицы T_s будет пропорционален первому столбцу $M - sE$.

Таким образом, матрица T будет матрицей QR -преобразования со сдвигом s , если ее первый столбец будет пропорционален первому столбцу матрицы $M - sE$. А так как $T = T_2 T_3 \dots T_n$, окончательно приходим к выводу, что T_2 в (IV.9) необходимо выбрать так, чтобы ее первый столбец был пропорционален $M - sE$. В этом случае преобразование (IV.9) будет эквивалентно QR -преобразованию со сдвигом s для матрицы M . Параметр сдвига s выбирается равным собственному значению нижнего минора матрицы M ,

$$\begin{bmatrix} m_{n-1,n-1} & m_{n-1,n} \\ m_{n,n-1} & m_{n,n} \end{bmatrix},$$

которое ближе к $m_{n-1,n-1}$. При таком выборе параметра метод обладает глобальной и почти всегда кубической сходимостью [191].

Таким образом, в результате преобразования (IV.8), (IV.9) сингулярное разложение для матрицы $J^{(0)}$ будет иметь вид

$$J^{(0)} = GDH^T,$$

где G и H — ортогональные матрицы.

IV.2. РЕШЕНИЕ СИСТЕМ С ПРЯМОУГОЛЬНЫМИ И КВАДРАТНЫМИ ВЫРОЖДЕННЫМИ МАТРИЦАМИ

1. Численное псевдообращение матриц. Для численного псевдообращения действительной матрицы $A, m \geq n$, уравнения (IV.1) сначала так же, как это сделано в п. 4 параграфа IV.1, приведем матрицу к двухдиагональному виду

$$J^{(0)} = P^T A Q, \quad (IV.14)$$

а затем осуществим сингулярное разложение двухдиагональной матрицы $J^{(0)}$:

$$J^{(0)} = GDH^T \quad (IV.15)$$

(см. п. 5 параграфа IV.1). Таким образом, окончательно имеем

$$A = PGDH^T Q^T, \quad (IV.16)$$

или

$$A = U_c D V^T, \quad (IV.17)$$

где $U_c = PG$, $V = QH$, $D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r, \dots, \sigma_n)$.

Тогда псевдообратная матрица $A^\#$ может быть вычислена из представления

$$A^\# = V D^\# U_c^T. \quad (VI.18)$$

Здесь

$$D^\# = \text{diag}(\sigma_1^\#, \sigma_2^\#, \dots, \sigma_r^\#, \dots, \sigma_n^\#), \quad (IV.19)$$

$$\sigma_i^\# = \begin{cases} \frac{1}{\sigma_i} & \text{для } \sigma_i > 0, \quad i = 1, 2, \dots, r, \\ 0 & \text{для } \sigma_i = 0, \quad i = r+1, \dots, n. \end{cases}$$

Пример 4. Вычислить на ЭВМ матрицу, псевдообратную матрице

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 3 & 3 \end{bmatrix}$$

примера 1. (Сравнить полученные результаты обоих примеров).

В результате сингулярного разложения исходной матрицы на ЭВМ получаем

$$U_c = \begin{bmatrix} -0,408\,247\,9 & 0,707\,107\,1 \\ -0,408\,247\,9 & -0,707\,106\,7 \\ -0,816\,496\,7 & -0,119\,209\,3 \cdot 10^{-6} \end{bmatrix},$$

$$D = \begin{bmatrix} 0,519\,615\,1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$V = \begin{bmatrix} -0,707\,106\,9 & -0,707\,106\,9 \\ -0,707\,107\,6 & 0,707\,107\,6 \end{bmatrix}.$$

Таким образом,

$$A^\# = V D^\# U_c^T = \begin{bmatrix} 0,555\,555\,8 & -0,444\,444\,5 & 0,111\,111\,0 \\ -0,444\,445\,2 & 0,555\,556\,1 & 0,111\,111\,3 \end{bmatrix}. \quad (IV.20)$$

2. Решение системы линейных алгебраических уравнений методом сингулярного разложения матрицы. Для системы линейных алгебраических уравнений (IV.1) с действительной прямоугольной или квадратной матрицей независимо от того, совместна система или нет, можно получить обобщенное решение задачи. Для вычисления нормального

псевдорешения можно использовать псевдообратную матрицу $A^\#$, т. е. определить нормальное псевдорешение по формуле

$$x^H = A^\# b = V D^\# U^T b. \quad (IV.21)$$

Пример 5. С помощью сингулярного разложения матрицы найти нормальное псевдорешение системы линейных алгебраических уравнений

$$\begin{aligned} 2x_1 - x_2 + \sqrt{2}x_3 &= 5 + 7\sqrt{2}, \\ 3x_1 + 2x_2 - 3x_3 &= -24, \end{aligned} \quad (IV.22)$$

$$3x_1 + \sqrt{2}x_2 - \frac{15}{7}x_3 = -13 - 3\sqrt{2}.$$

Решение. На ЭВМ МИР-2 при длине машинного слова 10 десятичных знаков в результате сингулярного разложения вырожденной матрицы ($\det A = 0$) данной системы получаем

$$\begin{aligned} U &= \begin{bmatrix} -0,026\,864\,273\,1 & 0,981\,525\,483 & 0,189\,436\,096\,8 \\ -7,648\,820\,599\,8 & -1,421\,939\,607 & 0,628\,280\,440 \\ 0,643\,609\,932\,2 & 0,128\,017\,973\,8 & 0,754\,571\,171\,0 \end{bmatrix}, \\ D &= \begin{bmatrix} 6,111\,777\,767 & 0 & 0 \\ 0 & 2,690\,354\,841 & 0 \\ 0 & 0 & 1,489\,553\,318 \cdot 10^{-18} \end{bmatrix}, \\ V^T &= \begin{bmatrix} -0,700\,157\,087\,7 & -0,394\,828\,122\,2 & 0,594\,887\,222\,2 \\ 0,713\,854\,907\,5 & -0,403\,243\,705\,0 & 0,572\,543\,174\,8 \\ 0,013\,828\,380\,9 & 0,825\,533\,322\,2 & 0,564\,183\,928 \end{bmatrix}. \end{aligned} \quad (IV.23)$$

В пределах неопределенности исходных данных (иррациональные числа и дроби, отличные от двоичных, непредставимы точно в ЭВМ) и ошибок округления в процессе счета реально вычисленное сингулярное число $\sigma_3 = 1,489\,553\,318 \cdot 10^{-18}$ следует рассматривать как нулевое, поэтому

$$D^\# = \begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \frac{1}{\sigma_2} & \\ & & 0 \end{bmatrix}.$$

Т а б л и ц а 6

Нормальное псевдорешение	Машинное нормальное псевдорешение, вычисленное с длиной машинного слова	
	10 десятичных знаков	30 десятичных знаков (округленное до 10 значащих цифр)
$\frac{-380 + 651\sqrt{2}}{552} = 0,979\,443\,8$	0,979 443 901	0,979 443 893
$\frac{-2274 - 42\sqrt{2}}{552} = -4,227\,168$	-4,227 168 427	-4,227 168 423
$\frac{2520 + 623\sqrt{2}}{552} = 6,161\,331$	6,161 331 606	6,161 371 611

Учитывая вид $D^\#$, вычисляем по формуле (IV.21) искомое нормальное псевдорешение системы (IV.22).

В табл. 6 приведено сравнение точного нормального псевдорешения системы (IV.23) с машинными нормальными псевдорешениями, полученными на ЭВМ МИР-2. Приведенные результаты демонстрируют высокую численную устойчивость алгоритма построения нормального псевдорешения, основанного на сингулярном разложении матрицы системы.

3. Численное решение однородных систем уравнений. Пусть необходимо решить систему линейных алгебраических уравнений

$$Ax = 0, \quad (IV.24)$$

где A — матрица с размерами $m \times n$, имеющая ранг $r(A)$.

Так как все решения системы (IV.24) удовлетворяют системе

$$A^T Ax = 0,$$

причем матрицы A и $A^T A$ имеют одинаковый ранг, то ортонормированные собственные векторы матрицы $A^T A$, отвечающие нулевым собственным значениям, представляют все линейно независимые решения системы (IV.24).

Таким образом, все решения однородной системы (IV.24) можно получить тоже с помощью сингулярного разложения матрицы

$$A = UDV^T.$$

Для этого достаточно отобрать все столбцы матрицы V , отвечающие нулевым сингулярным числам. Это и будут все линейно независимые решения системы (IV.24).

Пример 6. Определить на ЭВМ нетривиальное решение системы линейных алгебраических уравнений

$$\begin{aligned} 3x_1 - 2x_2 + 4x_3 &= 0, \\ 2x_1 - 3x_2 + 5x_3 &= 0, \end{aligned} \quad (IV.25)$$

ранг которой равен 2.

Решение. Применяя к матрице (IV.25) сингулярное разложение, получаем на ЭВМ МИР-2 с длиной машинного слова 10 десятичных знаков следующее сингулярное разложение матрицы системы (IV.25):

$$\begin{aligned} \begin{bmatrix} 3 & -2 & 4 \\ 2 & -3 & 5 \end{bmatrix} &= \begin{bmatrix} -0,656\,027\,883\,2 & 0,754\,736\,654\,5 \\ -0,754\,736\,652\,4 & -0,656\,027\,886\,3 \end{bmatrix} \times \\ &\times \begin{bmatrix} 8,112\,635\,649 & 0 & 0 \\ 0 & 1,088 & 642 & 605 & 0 \end{bmatrix} \times \\ &\times \begin{bmatrix} -0,428\,659\,329\,4 & 0,440\,826\,615\,1 & -0,788\,620\,995\,1 \\ 0,874\,625\,135\,4 & 0,421\,268\,045\,2 & -0,239\,925\,208\,6 \\ 0,226\,455\,407\,8 & -0,792\,593\,922\,4 & -0,566\,138\,516\,4 \end{bmatrix}. \end{aligned}$$

Таким образом, решение системы (IV.25), полученное на ЭВМ, имеет вид

$$x = \begin{bmatrix} 0,226\,455\,407\,8 \\ -0,792\,593\,922\,4 \\ -0,566\,138\,516\,4 \end{bmatrix},$$

и точное решение этой задачи, как нетрудно убедиться, будет

$$x = \frac{1}{\sqrt{78}} \begin{bmatrix} 2 \\ -7 \\ -5 \end{bmatrix} \approx \begin{bmatrix} 0,226\ 455\ 406 \\ -0,792\ 593\ 923 \\ -0,566\ 138\ 517 \end{bmatrix}.$$

4. Метод регуляризации А. Н. Тихонова. В работах [91, 93] рассмотрен метод нахождения нормального псевдорешения x^H системы линейных алгебраических уравнений (IV.1) с действительной прямоугольной или квадратной вырожденной матрицей, для которой известны оценки погрешности в исходных данных

$$\|A - \tilde{A}\| \leq \varepsilon_A \leq \delta, \quad \|b - \tilde{b}\| \leq \varepsilon_b \leq \delta \quad (\text{IV.26})$$

независимо от того, совместна система или нет. Доказана принципиальная возможность построения такого решения x_α всегда совместной системы

$$(A^T A + \alpha E) x_\alpha = A^T b, \quad \alpha > 0, \quad (\text{IV.27})$$

с симметричной и положительно определенной матрицей $A^T A + \alpha E$, которое будет как угодно близко к нормальному псевдорешению

$$\|x^H - x_\alpha\| \leq \varepsilon,$$

если точность исходных данных удовлетворяет некоторому условию $\delta \leq \delta_0(\varepsilon, \|x^H\|)$ и параметр α выбирается в зависимости от ε, δ в интервале

$$\frac{\delta^2}{\varepsilon(\delta)} \leq \alpha \leq \alpha_0(\delta),$$

где $\varepsilon(\delta)$ и $\alpha_0(\delta)$ — какие-либо функции, стремящиеся к нулю при $\delta \rightarrow 0$ и $\frac{\delta^2}{\varepsilon(\delta)} \leq \alpha_0(\delta)$.

Переход от решения заданной системы линейных алгебраических уравнений (IV.1) к решению системы уравнений (IV.27) получил название метода регуляризации А. Н. Тихонова. Исследование вычислительной устойчивости алгоритмов решения системы (IV.27) и минимизации объема вычислительной работы при этом можно найти в [21].

Пример 7. Методом регуляризации А. Н. Тихонова решить систему линейных алгебраических уравнений (IV.22).

Решение. Вместо системы (IV.22) будем рассматривать регуляризованную систему

$$\begin{aligned} 22,000\ 1\ x_1^{(\alpha)} + (4 + 3\sqrt{2})\ x_2^{(\alpha)} - \left(\frac{108}{7} + 2\sqrt{2}\right) x_3^{(\alpha)} &= -101 + 5\sqrt{2}, \\ (4 + 3\sqrt{2})\ x_1^{(\alpha)} + 7,000\ 1\ x_2^{(\alpha)} - \left(6 + \frac{22}{7}\sqrt{2}\right) x_3^{(\alpha)} &= -59 - 20\sqrt{2}, \quad (\text{IV.28}) \\ -\left(\frac{108}{7} + 2\sqrt{2}\right) x_1^{(\alpha)} - \left(6 + \frac{22}{7}\sqrt{2}\right) x_2^{(\alpha)} + \left(\frac{764}{49} + 0,000\ 1\right) x_3^{(\alpha)} &= \\ &= \frac{797 + 80\sqrt{2}}{7}. \end{aligned}$$

Таблица 7

Нормальное псевдорешение (округленное до 10 значащих цифр)	Машинное решение			
	с 10 десятичными знаками		с 30 десятичными знаками	
	нерегуляризованной системы	регуляризованной системы	нерегуляризованной системы	регуляризованной системы
0,979 443 892 9	2,725 336 900	0,979 393 820	1,078 170 605	0,979 394 108
-4,227 168 422	100,0	-4,227 474 53	1,666 666 666	-4,227 130 465
6,161 331 610	77,392 003 59	6,161 126 568	10,189 281 716	6,161 277 290

Сравнение точного нормального решения системы (IV.22) с решением регуляризованной (IV.28) и решениями системы (IV.22), полученными на ЭВМ МИР-2 с длиной машинного слова 10 и 30 десятичных знаков, приведено в табл. 7. Система уравнений (IV.28) решалась методом LL^T -разложения.

Полученные результаты свидетельствуют о том, что метод регуляризации А. Н. Тихонова при соответствующем выборе параметра регуляризации обеспечивает численно устойчивый алгоритм нахождения нормального псевдорешения произвольной системы линейных алгебраических уравнений. В то же время обычные прямые методы, в частности метод квадратных корней, реализованный в матобеспечении ЭВМ МИР-2, оказываются непригодными для построения нормального псевдорешения: в зависимости от накопления ошибок округления в процессе вычислений (и погрешностей ввода чисел в ЭВМ) они вычисляют некоторое случайное решение, минимизирующее вектор невязки.

Вопрос об эффективном выборе параметра регуляризации относится к числу трудных практических вопросов. В ряде работ предлагались различные методы определения α . По этим методам были разработаны численные алгоритмы решения регуляризованных систем (см., например, [21]).

Для практического нахождения α рекомендуется следующий прием [36]. Вначале для некоторого α находят x_α , вычисляют невязку $g_\alpha = Ax_\alpha - b$ и сравнивают ее по норме с известной погрешностью правых частей ε_b и произведением погрешности коэффициентов матрицы ε_A на x_α . Если значение α слишком велико, то невязка заметно больше этих величин, если слишком мало, то невязка заметно меньше. Проводят серию расчетов с различными α ; оптимальным считают тот, в котором $\|g_\alpha\| \approx \varepsilon_b + \varepsilon_A \|x_\alpha\|$.

Отметим, что метод регуляризации А. Н. Тихонова реализуется примерно в три раза быстрее, чем метод разложения по сингулярным числам. Однако он требует для своей реализации ЭВМ с большой длиной машинного слова, так как в противном случае при перемножении матрицы A на матрицу A^T можно получить на машине произведение $A^T A$, сильно отличающееся от истинного математического произведения этих матриц. Отметим, что при решении регуляризованной системы (IV.27) матрицу $A^T A$ не строят в явном виде.

5. Нормализованный процесс и его применение к решению систем с произвольными прямоугольными матрицами. Сущность нормализованного процесса [53, 105] состоит в том, что прямоугольная матрица A ранга k посредством правосторонних умножений на ортогональные матрицы (например, элементарные матрицы отражения, см. п. 8 параграфа I.1) приводится к левой трапецевидной матрице L . При этом

необходимо выполнять перестановку строк преобразуемой матрицы таким образом, чтобы для результирующей матрицы

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ l_{k1} & l_{k2} & \dots & l_{kk} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ l_{m1} & l_{m2} & \dots & l_{mk} & 0 & \dots & 0 \end{bmatrix} \quad (VI.29)$$

были справедливы неравенства

$$|l_{11}| \geq |l_{22}| \geq \dots \geq |l_{kk}|, \quad |l_{ij}| \leq |l_{jj}|, \quad i > j. \quad (IV.30)$$

Этим условиям легко удовлетворить. Действительно, поскольку при умножении матрицы справа на ортогональную матрицу сумма квадратов элементов каждой строки остается неизменной, для обеспечения (IV.30) достаточно придерживаться следующего порядка выбора строк. Перед получением нулей в первой строке нужно изменить порядок строк у исходной матрицы A , так чтобы первой строкой оказалась строка с наибольшей суммой квадратов элементов. Далее, перед получением нулей во второй строке нужно поставить в качестве второй строки преобразуемой матрицы строку с наибольшей суммой квадратов элементов, не считая первой, и т. д. Очевидно, что такой способ выбора строк обеспечивает у матрицы выполнение условий (IV.30).

Трапециевидная матрица L связана с исходной A соотношением

$$L = PAQ, \quad (IV.31)$$

где P — результирующая матрица перестановок, Q — результирующая ортогональная матрица. Чтобы найти Q , нужно все последовательные элементарные матрицы отражений (каждая из которых обеспечивает образование нулей в соответствующей строке преобразуемой матрицы) перемножить. Отдельно подсчитывается результирующая матрица P .

Введем следующие обозначения: $L = [L_0 : 0]$, $Q = [Q_0 : Q_1]$, где L_0 — прямоугольная матрица с размерами $m \times k$, число столбцов которой равно рангу матрицы A ; Q_0 — прямоугольная матрица с размерами $n \times k$; Q_1 — матрица с размерами $n \times (n - k)$, для которой справедливо соотношение $AQ_1 = 0$.

Отметим, что ранги матриц L_0 и Q_0 равны k , т. е. рангу A . Далее, поскольку столбцы матрицы Q_1 линейно независимы, они являются базисом подпространства решений уравнения

$$Ay = 0.$$

В этих обозначениях справедливо разложение

$$PA = L_0 Q_0^T, \quad (IV.32)$$

которое называется нормализованным разложением матрицы A .

Отметим, что при реализации нормализованного процесса на ЭВМ элементы матрицы L будут получаться с определенной точностью ε .

Поэтому элемент $l_{k+1,k+1}$ следует рассматривать как нулевой, если выполняется неравенство

$$\left| \frac{l_{k+1,k+1}}{l_{kk}} \right| < \varepsilon. \quad (IV.33)$$

Выбор параметра ε на практике должен диктоваться точностью вычислений конкретной ЭВМ и точностью задания элементов матрицы. С точностью до выбранного ε нормализованный процесс позволяет определить ранг произвольной матрицы A и базис ее нуль-пространства ($AQ_1 = 0$).

Рассмотрим теперь применение нормализованного процесса к решению системы линейных алгебраических уравнений (IV.1). Учитывая нормализованное разложение матрицы A (IV.32), перейдем от системы (IV.1) к решению эквивалентной системы

$$L_0 z = g, \quad (IV.34)$$

где $z = Q_0^T x$, $g = Pb$. Определив z , искомое решение x можно вычислить по формуле $x = Q_0 z$.

Для отыскания обобщенного решения переопределенной системы (IV.34) с матрицей полного ранга можно поступить следующим образом. С помощью умножений слева на последовательность элементарных матриц отражения можно преобразовать систему (IV.34) к виду

$$Rz = f,$$

где

$$TL_0 \equiv R = \begin{bmatrix} R_0 \\ 0 \end{bmatrix}, \quad f = Tg = \begin{bmatrix} f_0 \\ f_1 \end{bmatrix}.$$

R_0 — правая треугольная матрица с размерами $k \times k$, T — результирующая матрица ортогональных преобразований. Решая систему

$$R_0 z = f_0,$$

можно найти вектор z , а затем вычислить $x_H = Q_0 z$. Заметим, что нормализованный процесс благодаря описанной выше перестановке строк обеспечивает высокую численную устойчивость.

Нормализованный процесс можно применить и для построения псевдообратных матриц. Как показано в работе [53], формула для вычисления псевдообратной матрицы имеет вид

$$A^* = Q_0 (L_0^T L_0)^{-1} L_0^T P.$$

Пример 8. Найти с помощью нормализованного процесса нормальное псевдорешение системы уравнений примера 2.

Решение. После выбора «максимальной» строки на первом шаге и соответствующей перестановки строк имеем

$$PA = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 3 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad Pb = \begin{bmatrix} 9 \\ 5 \\ 4 \end{bmatrix} = g.$$

Умножение справа матрицы PA на элементарную матрицу отражения (для получения нулей в первой строке) позволяет получить

$$PAQ = \begin{bmatrix} -3\sqrt{2} & 0 \\ -\frac{3}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{3}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = L,$$

где

$$Q = E - \frac{1}{\alpha} \mathbf{w} \mathbf{w}^T = -\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$\mathbf{w}^T = [3 + 3\sqrt{2}, 3], \quad \alpha = 9\sqrt{2}(\sqrt{2} + 1).$$

Теперь необходимо решить переопределенную систему

$$Lz = g.$$

После первого умножения матрицы L и вектора g на матрицу отражения $T_1 = E - \frac{1}{\beta_1} \mathbf{u}_1 \mathbf{u}_1^T$ ($\beta_1 = 27 + 9\sqrt{6}$; $\mathbf{u}_1^T = [-3(\sqrt{2} + \sqrt{3}), -\frac{3}{\sqrt{2}}, -\frac{3}{\sqrt{2}}]$)

получим систему

$$R_1 z = f_1,$$

где

$$R_1 \equiv TL = \begin{bmatrix} 3\sqrt{3} & 0 \\ 0 & \frac{1}{\sqrt{2}} \\ 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad f_1 \equiv T_1 g = \begin{bmatrix} -\frac{27}{\sqrt{6}} \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}, \quad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}.$$

После второго шага левого умножения матрицы L и вектора g на матрицу отражения $T_2 = E - \frac{1}{\beta_2} \mathbf{u}_2 \mathbf{u}_2^T$, $\beta_2 = (1 + \frac{1}{\sqrt{2}})$, $\mathbf{u}_2^T = [0, \frac{1}{\sqrt{2}} + 1, -\frac{1}{\sqrt{2}}]$ получим систему

$$R_2 z = f_2,$$

где

$$R_2 \equiv T_2 T_1 L = \begin{bmatrix} 3\sqrt{3} & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix}, \quad f_2 \equiv T_2 T_1 g = \begin{bmatrix} -\frac{27}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}.$$

Отсюда непосредственно находят $z_1 = -\frac{3}{\sqrt{2}}$, $z_2 = \frac{1}{\sqrt{2}}$. Вычисления искомого решения $\mathbf{x}_H$ осуществляют по формуле

$$\mathbf{x}_H = Qz = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} -\frac{3}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

6. Сдвиг спектра матрицы для решения систем с квадратными положительно полуопределенными матрицами. В случае, когда в (IV.1) A — действительная положительно полуопределенная матрица, заданная точно, возможен и несколько другой подход к получению одного из обобщенных решений системы (IV.1), а именно: путем сдвига спектра матрицы A [171].

В качестве приближения к некоторому обобщенному решению принимают единственное решение системы

$$(A + \zeta E) \mathbf{x}_\zeta = \mathbf{b}, \quad \zeta > 0. \quad (IV.35)$$

Справедлива следующая оценка погрешности вектора $\mathbf{x}_\zeta$, рассматриваемого как приближение к некоторому решению $\mathbf{x}^*$ системы (IV.3):

$$\|\mathbf{x}_\zeta - \mathbf{x}^*\| \leq \frac{\zeta}{\lambda_{m+1}} \|\bar{\mathbf{x}}^*\| \leq \frac{\zeta}{\lambda_{m+1}} \|\mathbf{x}^*\|, \quad \|\mathbf{x}_\zeta - \mathbf{x}^*\| \leq \frac{\zeta}{\lambda_{m+1}} \|\bar{\mathbf{x}}_\zeta\|, \quad (IV.36)$$

где $\bar{\mathbf{x}}^*$ (и соответственно $\bar{\mathbf{x}}_\zeta$) — проекция вектора $\mathbf{x}^*$ (и $\mathbf{x}_\zeta$) на собственное подпространство, отвечающее ненулевым собственным значениям матрицы A ; λ_{m+1} — первое, отличное от нулевого, собственное значение. В оценках (IV.36) предполагается, что $\mathbf{x}_\zeta$ и $\mathbf{x}^*$ имеют одинаковые проекции на подпространство, образованное собственными векторами, отвечающими нулевому собственному значению матрицы A . Отметим, что в силу выбора решения $\mathbf{x}^*$ справедливо равенство

$$\mathbf{x}_\zeta - \mathbf{x}^* \equiv \bar{\mathbf{x}}_\zeta - \bar{\mathbf{x}}^*.$$

Если ζ достаточно велико, то система (IV.35) будет достаточно хорошо обусловлена, так как

$$\text{cond}(A + \zeta E) = \frac{\lambda_n + \zeta}{\zeta},$$

где λ_n — максимальное собственное значение матрицы A , и вычисление ее решения будет хорошо реализовываться на ЭВМ. Однако полученное решение может сильно отличаться от искомого обобщенного решения задачи. Если ζ очень мало, то система (IV.35) плохо обусловлена, и хотя теоретически ее решение хорошо приближает одно из обобщенных решений, ошибки машинной реализации могут сделать машинное решение задачи неприемлемым для практики.

Таким образом, при практическом выборе параметра ζ необходимо, с одной стороны, учитывать требование близости $\mathbf{x}_\zeta$ к $\mathbf{x}^*$ и тем самым ограничивать ζ сверху, а с другой — учитывать обусловленность матрицы системы, длину машинного слова и ограничивать ζ снизу.

Для решения системы линейных алгебраических уравнений (IV.35) с симметричной и положительно определенной матрицей целесообразно применять метод LL^T -разложения.

Пример 9. Найти одно из обобщенных решений системы линейных алгебраических уравнений

$$A\mathbf{x} = \mathbf{b} \quad (IV.37)$$

с симметричной и положительно полуопределенной матрицей A десятого порядка

$$A = \begin{bmatrix} 1 & -1 & & & & & & & & \\ -1 & 2 & -1 & & & & & & & \\ & -1 & 2 & -1 & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & -1 & 2 & -1 & & & \\ & & & & -1 & & 1 & & & \end{bmatrix}$$

и правой частью

$$b^T = [4,82, -4,87, -1,13, -1,69, 4,18, 1,11, -2,28, -0,1, 0,08, -0,02]$$

с относительной погрешностью порядка 0,6 %. Отметим, что правая часть системы (IV.37) вследствие погрешности в задании исходных данных не удовлетворяет классическим условиям разрешимости, причем евклидова норма погрешности вектора b не превышает $\varepsilon_b = 0,01 \sqrt{10}$. Для матрицы A первое отличное от нуля собственное значение $\lambda_{m+1} = \lambda_2 = 0,097\ 887\ 5$, а максимальное собственное значение $\lambda_n = \lambda_{10} = 3,902\ 111\ 2$.

Решение. Для решения системы (IV.37) применим метод сдвига спектра. Учитывая оценку (IV.36) и значение λ_2 , требуемую точность (0,6 %) можно гарантировать уже при $\zeta = 0,5 \cdot 10^{-3}$ (истинная относительная погрешность 0,511 %). Тогда количество s_1 точных десятичных знаков у точной проекции $\bar{x}_\zeta$ по отношению к точному решению $\bar{x}^*$ будет не меньше, чем значение $\lg \frac{\lambda_2}{\zeta}$, округленное до ближайшего целого¹⁷, т. е. в рассматриваемом случае $s_1 = 2$. Обусловленность системы (IV.35) при выбранном ζ окажется равной $\text{cond}(A + \zeta E) \approx \frac{3,9}{0,5} 10^3 \approx 7,8 \cdot 10^3 \approx 10^4$. Учитывая формулу (I.95), имеем

$$\text{cond}(A + \zeta E) \approx \frac{1}{\varepsilon} 10^{-\eta},$$

где η — количество десятичных разрядов вычисленного на ЭВМ решения, не испорченных ошибками округления, а ε — единичная ошибка округления данной ЭВМ, измеренная в относительных единицах: $1,0 + \varepsilon = 1,0$. В данном случае при длине машинного слова t имеем $10^4 \approx 10^{t-\eta}$, т. е. $\eta \approx t - 4$. Отсюда ясно, что длина машинного слова на ЭВМ МИР-2, выбранная для вычисления решения системы (IV.35) при $\zeta = 0,5 \cdot 10^{-3}$, должна быть больше четырех. Для окончательного решения вопроса о выборе ζ и t необходимо учесть еще следующее обстоятельство. В точном решении десятичные цифры, содержащие проекцию $\bar{x}_\zeta$ (именно эта проекция и является полезной информацией в решении x_ζ), стоят не далее, чем после s первых десятичных цифр, где s определяется значением

$$s = \lg \frac{\|x_\zeta\|}{\|\bar{x}_\zeta\|},$$

округленным до ближайшего целого. Чтобы предохранить от искажения ошибками округления хотя бы k верных десятичных знаков проекции $\bar{x}_\zeta$ у вычисленного решения x_ζ , необходимо гарантировать выполнение неравенства

$$\eta \geq s + k, \quad (IV.38)$$

где $1 \leq k \leq s_1$.

¹⁷ Здесь и в дальнейшем используется известный факт связи значения относительной погрешности ε приближенного числа и количества s_1 верных десятичных разрядов у него, а именно: $s_1 = -\lg \varepsilon$.

Т а б л и ц а 8

$\bar{x}^*$	x^*	x_ζ		Δ
3,27	23,27	23,266 063	0,12	0,017
-1,54	18,46	18,457 696	0,14	0,012
-1,47	18,53	18,528 558	0,097	0,008
-0,26	19,74	19,738 686	0,505	0,007
2,65	22,65	22,648 686	0,05	0,006
1,39	21,39	21,390 013	0,001	$0,6 \cdot 10^{-4}$
-0,97	19,03	19,032 031	0,21	0,011
-1,04	18,96	18,963 563	0,34	0,018
-1,00	19,00	19,004 577	0,46	0,024
-1,03	18,97	18,975 090	0,49	0,027

Соотношение (IV.38), равносильное

$$10^\eta \geq 10^k \frac{\|x_\zeta\|}{\|\bar{x}_\zeta\|}, \quad 10^\eta \approx \frac{1}{\varepsilon} \frac{1}{\text{cond}(A + \zeta E)},$$

будет заведомо выполняться, если справедливо неравенство

$$\frac{1}{\varepsilon} \frac{1}{\text{cond}(A + \zeta E)} \geq 10^k \left[1 + \left(\frac{\lambda_n}{\zeta} \frac{\varepsilon_b}{\|b\|} \right)^2 \right]^{1/2}, \quad (IV.39)$$

так как

$$\frac{\|x_\zeta\|}{\|\bar{x}_\zeta\|} \leq \left[1 + \left(\frac{\lambda_n}{\zeta} \frac{\varepsilon_b}{\|b\|} \right)^2 \right]^{1/2}.$$

Отметим, что неравенство (IV.39) можно переписать в виде

$$\frac{\zeta}{\lambda_n} \geq \eta 10^k \left[1 + \left(\frac{\lambda_n}{\zeta} \frac{\varepsilon_b}{\|b\|} \right)^2 \right]^{1/2}. \quad (IV.40)$$

В рассматриваемом примере неравенство (IV.39) при выбранном $\zeta = 0,5 \cdot 10^{-3}$, $1 \leq k \leq 2$, $\varepsilon = 10^{-t}$ принимает вид $10^{t-4} \geq 10^k \cdot 28,5$. Отсюда следует, что для сохранения двух верных цифр полезной информации, т. е. при $k = 2$, разрядность ЭВМ должна быть не менее семи. На основании проведенных рассуждений выберем для решения задачи $t = 8$. Значения вычисленного x_ζ и соответствующего точного обобщенного решения x^* , а также $\bar{x}^*$, приведены в табл. 8, где даны также покомпонентно следующие относительные погрешности в процентах для вычисленного вектора:

$$\Delta = \frac{|x_\zeta^{(i)} - x_i^*|}{|x_i^*|} 100 \%, \quad \delta = \frac{|x_\zeta^{(i)} - \bar{x}_i^*|}{|\bar{x}_i^*|} 100 \%.$$

Рассмотрение табл. 8 позволяет сделать вывод о совпадении теоретических предсказаний и результатов выполненного расчета.

IV.3. ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ С НЕЕДИНСТВЕННЫМ РЕШЕНИЕМ И НЕСОВМЕСТНЫХ СИСТЕМ С СИММЕТРИЧНЫМИ И ПОЛОЖИТЕЛЬНО ПОЛУОПРЕДЕЛЕННЫМИ МАТРИЦАМИ

1. Методы решения систем с неединственным решением. Рассмотрим систему линейных алгебраических уравнений

$$Ay = f \quad (IV.41)$$

с положительно полуопределенной матрицей $A = A^T$, $\det A = 0$, причем вначале будем полагать, что решение системы (IV.41) существует, хотя и будет не единственным.

Для решения системы (IV.41) можно использовать итерационные процессы, записанные в канонической форме (II.53) (см. п. 4 параграфа II.2):

$$B \frac{y^{(k+1)} - y^{(k)}}{\tau} + Ay^{(k)} = f, \quad y^{(0)} = y_0, \quad (IV.42)$$

где B — симметричная и положительно определенная матрица, $BA = AB$, τ — итерационный параметр.

Итерационный процесс, записанный в виде (IV.42), при

$$\tau = \frac{2}{\gamma_1 + \gamma_2}, \quad (IV.43)$$

где $\gamma_1 > 0$ — оценка снизу наименьшего собственного значения, отличного от нулевого, $\gamma_2 \geq \gamma_1$ — оценка сверху максимального собственного значения матрицы $C = B^{-\frac{1}{2}}AB^{-\frac{1}{2}}$, сходится со скоростью геометрической прогрессии, знаменатель которой

$$\rho = \frac{\gamma_2 - \gamma_1}{\gamma_2 + \gamma_1} \quad (IV.44)$$

к одному из решений системы (IV.41), определяемому выбором начального приближения. В случае нулевого начального приближения итерационный процесс будет сходиться к нормальному псевдорешению [65].

Итерационную схему (IV.36) можно представить в виде

$$\frac{x^{(k+1)} - x^{(k)}}{\tau} + Cx^{(k)} = g, \quad x^{(0)} = B^{\frac{1}{2}}y_0, \quad (IV.45)$$

где $x = B^{\frac{1}{2}}y$, $g = B^{-\frac{1}{2}}f$, $C = B^{-\frac{1}{2}}AB^{-\frac{1}{2}}$. Такое представление итерационной схемы возможно, так как B — положительно определенная матрица, и поэтому существуют положительно определенные $B^{\frac{1}{2}}$ и $B^{-\frac{1}{2}}$ матрицы.

Введем в рассмотрение погрешность $z^{(k)} = x^{(k)} - x^H$. Здесь x^H — нормальное псевдорешение системы

$$Cx = g, \quad (IV.46)$$

а $x^{(k)}$ — приближение к решению системы уравнений (IV.46) на k -й итерации ($k = 0, 1, 2, \dots$). Тогда итерационную схему (IV.45) относительно погрешности $z^{(k)}$ можно записать следующим образом:

$$\frac{z^{(k+1)} - z^{(k)}}{\tau} + Cz^{(k)} = 0, \quad z^{(0)} = x^{(0)} - x^H,$$

или

$$z^{(k+1)} = (E - \tau C)z^{(k)}, \quad z^{(0)} = x^{(0)} - x^H, \quad (IV.47)$$

где E — единичная матрица.

Так как C — симметричная и положительно полуопределенная матрица, то она имеет полную ортогональную систему собственных векторов $\{\psi_j\}$, $j = 1, 2, \dots, n$, соответствующих собственным значениям

$$0 = \mu_1 = \dots = \mu_m < \gamma_1 \leq \mu_{m+1} \leq \mu_{m+2} \leq \dots \leq \mu_n \leq \gamma_2.$$

Подпространство, натянутое на собственные векторы $\{\psi_j\}$, $j = 1, 2, \dots, m$, соответствующие нулевым собственным значениям, обозначим через S_0^c , подпространство, порожденное собственными векторами $\{\psi_j\}$, $j = m+1, \dots, n$, соответствующими ненулевым собственным значениям, — через S_1^c .

Представим $z^{(k)}$ в виде разложения по системе собственных векторов

$$z^{(k)} = \sum_{j=1}^n c_j^{(k)} \psi_j, \quad k = 0, 1, \dots \quad (IV.48)$$

и подставим его в (IV.47), в результате получим

$$\sum_{j=1}^n c_j^{(k+1)} \psi_j = \sum_{j=1}^n (1 - \tau \mu_j) c_j^{(k)} \psi_j. \quad (IV.49)$$

Учитывая ортогональность системы векторов $\{\psi_j\}$, вместо (IV.49) можно записать

$$c_j^{(k+1)} = (1 - \tau \mu_j) c_j^{(k)} = (1 - \tau \mu_j)^{k+1} c_j^{(0)}, \quad j = m+1, m+2, \dots, n, \quad (IV.50)$$

$$c_j^{(k+1)} = c_j^{(k)} = c_j^{(0)}, \quad j = 1, 2, \dots, m. \quad (IV.51)$$

Из формулы (IV.51) следует, что проекции $c_j^{(k)}$ на подпространство S_0^c остаются неизменными и совпадают с проекциями начальной погрешности $z^{(0)}$.

Так как решения системы (IV.46) могут иметь произвольные проекции на подпространство S_0^c , то для сходимости процесса итераций достаточно, чтобы

$$\lim_{k \rightarrow \infty} |c_j^{(k)}| = 0, \quad j = m+1, m+2, \dots, n. \quad (IV.52)$$

Рассуждая так же, как в п. 4 параграфа II.2, можно доказать, что τ целесообразно вычислять по формуле (IV.43).

Представим $z^{(k)}$ в виде суммы двух проекций, одна из которых $\tilde{z}^{(k)}$ есть проекция на подпространство S_0^c , а вторая $\bar{z}^{(k)}$ — на подпространство S_1^c , т. е.

$$z = \tilde{z} + \bar{z} = \sum_{j=1}^m c_j \psi_j + \sum_{j=m+1}^n c_j \psi_j,$$

и введем в рассмотрение евклидову норму векторов, т. е. будем полагать

$$\|z\| = \sqrt{(z, z)}.$$

На основании разложения $\bar{z}$ по собственным векторам и равенства (IV.50) можно записать

$$\|\bar{z}^{(k)}\| = \left\| \sum_{i=m+1}^n c_i^{(k)} \psi_i \right\| \leq \rho \|\bar{z}^{(k-1)}\| \leq \rho^k \|\bar{z}^{(0)}\|.$$

Таким образом, уменьшение проекций погрешности на подпространство S_1^c происходит со скоростью геометрической прогрессии, знаменатель которой равен ρ . Учитывая, что проекции погрешности на подпространство S_0^c остаются неизменными и определяются начальным приближением к решению, получаем, что рассматриваемый итерационный процесс сходится к одному из решений системы (IV.41). Нетрудно убедиться, что в случае нулевого начального приближения построенный нами итерационный процесс будет сходиться к нормальному псевдорешению.

Иногда затруднительно получить оценку снизу γ_1 первого отличного от нулевого собственного значения матрицы $C = B^{-\frac{1}{2}} A B^{-\frac{1}{2}}$. Тогда вместо оценки снизу γ_2 можно взять любое число $\gamma_1 = \gamma_1 + \alpha < \gamma_2$, где $0 \leq \alpha \leq \mu_{m+1} \leq \mu_n$ [121]. В этом случае вместо параметра τ в (IV.43) можно использовать параметр

$$\bar{\tau} = \frac{2}{\gamma_1 + \gamma_2} = \frac{2}{\gamma_1 + \alpha + \gamma_2} = \frac{2}{\gamma_1 + \gamma_2}.$$

Повторяя предыдущие рассуждения, можно показать, что если матрица A положительно полуопределена, матрица B положительно определена и вместо итерационного параметра τ используется параметр $\bar{\tau}$, то итерационный процесс, записанный в канонической форме (IV.42), сходится со скоростью геометрической прогрессии, знаменатель которой

$$\bar{\rho} = \frac{\gamma_2 - \gamma_1}{\gamma_2 + \gamma_1},$$

к одному из решений системы (IV.41), определяемому выбором начального приближения. Естественно, что в этом случае

$$\bar{\rho} > \rho,$$

и скорость сходимости итерационного процесса уменьшается.

2. Итерационные методы получения обобщенного решения одного класса несовместных систем линейных алгебраических уравнений. Пусть требуется найти обобщенное решение системы уравнений

$$Ay = f \quad (IV.53)$$

с симметричной положительно полуопределенной матрицей A , $\det A = 0$ и правой частью f , не удовлетворяющей классическому условию разрешимости.

Для получения обобщенного решения (см. п. 1 параграфа IV.1) несовместной системы (IV.53) в случае $BA = AB$ можно использовать итерационный процесс, записанный в канонической форме (IV.42).

Этот процесс можно заканчивать при выполнении неравенства

$$\|r^{(k)} - r^{(k+1)}\| \leq \varepsilon, \quad r^{(k)} = Ay^{(k)} - f, \quad (IV.54)$$

где ε — заданное малое число. Затем приближение к обобщенному решению u вычисляют по формуле

$$u^{(k)} = y^{(k)} - k(y^{(k+1)} - y^{(k)}). \quad (IV.55)$$

При этом справедлива оценка

$$\|u^{(k)} - u\| \leq \frac{(1 + k\tau\gamma_2)}{\tau\gamma_1^2\nu} \varepsilon, \quad (IV.56)$$

где ν — минимальное собственное значение матрицы B [65]. Итерационный процесс по схеме (IV.42) в случае $BA = AB$ применительно к несовместной системе (IV.53) при τ , вычисляемом по формуле (IV.43), дает последовательность векторов $\{y^{(k)}\}$, для которых справедливо соотношение

$$\lim_{k \rightarrow \infty} \|r^{(k+1)} - r^{(k)}\| = 0. \quad (IV.57)$$

Покажем это. Итерационный процесс (IV.42) представим в виде

$$\frac{x^{(k+1)} - x^{(k)}}{\tau} + Cx^{(k)} = g, \quad x^{(0)} = B^{\frac{1}{2}} y_0, \quad (IV.58)$$

где

$$x = B^{\frac{1}{2}} y, \quad g = B^{-\frac{1}{2}} f = \tilde{g} + \bar{g}, \quad \tilde{g} \in S_0^c, \quad \bar{g} \in S_1^c.$$

Тогда для погрешности $z^{(k)} = x^{(k)} - x^H$, где x^H — нормальное псевдорешение совместной системы

$$Cx = \bar{g}, \quad x \in S_1^c, \quad (IV.59)$$

имеем

$$z^{(k+1)} = (E - \tau C) z^{(k)} + \tau \tilde{g}, \quad z^{(0)} = x^{(0)} - x^H. \quad (IV.60)$$

Подставляя разложение

$$z^{(k)} = \sum_{i=1}^n c_i^{(k)} \psi_i, \quad \tilde{g} = \sum_{i=1}^m \beta_i \psi_i$$

в (IV.60), получаем

$$c_i^{(k+1)} = (1 - \tau\mu_i) c_i^{(k)} = (1 - \tau\mu_i)^{(k+1)} c_i^{(0)}, \quad j = m+1, m+2, \dots, n, \quad (IV.61)$$

$$c_i^{(k+1)} = c_i^{(k)} + \tau\beta_j, \quad j = 1, 2, \dots, m, \quad (IV.62)$$

где $\{\psi_j\}$ — полная ортогональная система собственных векторов матриц C и A (в силу выбора B , удовлетворяющего равенству $AB = BA$), μ_j — собственные значения матрицы C .

Ранее, в п. 1 настоящего параграфа было показано, что процесс уменьшения проекций погрешности на подпространство S_1^c происходит

со скоростью геометрической прогрессии, знаменатель которой равен

$$\rho = \frac{\gamma_2 - \gamma_1}{\gamma_2 + \gamma_1}.$$

Из (IV.62) можно записать

$$c_j^{(k+1)} = c_j^{(0)} + (k+1) \tau \beta_j, \quad j = 1, 2, \dots, m,$$

т. е.

$$\tilde{\mathbf{z}}^{(k)} = \tilde{\mathbf{z}}^{(0)} + k \tau \tilde{\mathbf{g}}. \quad (\text{IV.63})$$

Из формулы (IV.63) следует расходимость рассматриваемого нами итерационного процесса ($\|\mathbf{z}^{(k)}\| \rightarrow \infty$ при $k \rightarrow \infty$). Но нам нужно доказать, что выполняется (IV.57). Для этого оценим выражение

$$\|B^{-\frac{1}{2}} \mathbf{r}^{(k+1)} - B^{-\frac{1}{2}} \mathbf{r}^{(k)}\|.$$

$$\|B^{-\frac{1}{2}} \mathbf{r}^{(k+1)} - B^{-\frac{1}{2}} \mathbf{r}^{(k)}\| = \|B^{-\frac{1}{2}} A \mathbf{y}^{k+1} - B^{-\frac{1}{2}} \mathbf{f} - B^{-\frac{1}{2}} A \mathbf{y}^{(k)} + B^{-\frac{1}{2}} \mathbf{f}\| =$$

$$= \|B^{-\frac{1}{2}} A B^{-\frac{1}{2}} \mathbf{x}^{(k+1)} - B^{-\frac{1}{2}} A B^{-\frac{1}{2}} \mathbf{x}^{(k)}\| = \|C \mathbf{x}^{(k+1)} - C \mathbf{x}^{(k)}\| =$$

$$= \|C(\tilde{\mathbf{x}}^{(k+1)} + \bar{\mathbf{x}}^{(k+1)}) - C(\tilde{\mathbf{x}}^{(k)} + \bar{\mathbf{x}}^{(k)})\| = \|C\tilde{\mathbf{x}}^{(k+1)} - \bar{\mathbf{g}} - C\bar{\mathbf{x}}^{(k)} + \bar{\mathbf{g}} +$$

$$+ C(\tilde{\mathbf{x}}^{(k+1)} - \tilde{\mathbf{x}}^{(k)})\| = \|C\tilde{\mathbf{x}}^{(k+1)} - C\mathbf{x}^H - (C\bar{\mathbf{x}}^{(k)} - C\mathbf{x}^H) +$$

$$+ C(\tilde{\mathbf{x}}^{(0)} + (k+1)\tau\tilde{\mathbf{g}} - \tilde{\mathbf{x}}^{(0)} - k\tau\tilde{\mathbf{g}})\| = \|C(\tilde{\mathbf{x}}^{(k+1)} - \mathbf{x}^H) - C(\tilde{\mathbf{x}}^{(k)} - \mathbf{x}^H) +$$

$$+ \tau C\tilde{\mathbf{g}}\| = \|C(\tilde{\mathbf{z}}^{(k+1)} - \tilde{\mathbf{z}}^{(k)}) + \tau C \sum_{j=1}^m \beta_j \boldsymbol{\psi}_j\| =$$

$$= \|C(\tilde{\mathbf{z}}^{(k+1)} - \tilde{\mathbf{z}}^{(k)}) + \tau \sum_{j=1}^m \mu_j \beta_j \boldsymbol{\psi}_j\| = \|C(\tilde{\mathbf{z}}^{(k+1)} - \tilde{\mathbf{z}}^{(k)})\|,$$

так как $\mu_j = 0, j = 1, 2, \dots, m$. Но

$$\|C(\tilde{\mathbf{z}}^{(k+1)} - \tilde{\mathbf{z}}^{(k)})\| \leq \gamma_2 \|\tilde{\mathbf{z}}^{(k+1)} - \tilde{\mathbf{z}}^{(k)}\| \leq \gamma_2 (\|\tilde{\mathbf{z}}^{(k+1)}\| + \|\tilde{\mathbf{z}}^{(k)}\|) \leq$$

$$\leq \gamma_2 (\rho^{k+1} \|\tilde{\mathbf{z}}^{(0)}\| + \rho^k \|\tilde{\mathbf{z}}^{(0)}\|) \leq \gamma_2 (1 + \rho) \rho^k \|\tilde{\mathbf{z}}^{(0)}\|.$$

Таким образом,

$$\|B^{-\frac{1}{2}} \mathbf{r}^{(k+1)} - B^{-\frac{1}{2}} \mathbf{r}^{(k)}\| \leq \gamma_2 (1 + \rho) \rho^k \|\tilde{\mathbf{z}}^{(0)}\|. \quad (\text{IV.64})$$

Так как $B^{-\frac{1}{2}}$ положительно определенная матрица, то

$$\kappa \|\mathbf{r}^{(k+1)} - \mathbf{r}^{(k)}\| \leq \|B^{-\frac{1}{2}} \mathbf{r}^{(k+1)} - B^{-\frac{1}{2}} \mathbf{r}^{(k)}\|,$$

где κ — минимальное собственное значение матрицы $B^{-\frac{1}{2}}$. Отсюда согласно (IV.64) следует

$$\lim_{k \rightarrow \infty} \|\mathbf{r}^{(k+1)} - \mathbf{r}^{(k)}\| = 0.$$

Пусть $\{\mathbf{y}^{(k)}\}$ определяется по итерационной схеме (IV.42), тогда последовательность векторов (IV.55) сходится к обобщенному решению $\mathbf{u}$, имеющему одинаковую с $\mathbf{y}_0$ проекцию на подпространство S_0^c (S_0^A), т. е.

$$\lim_{k \rightarrow \infty} \|\mathbf{u}^{(k)} - \mathbf{u}\| = 0.$$

Так как $B^{-\frac{1}{2}} \mathbf{u} = \mathbf{x}^H + \tilde{\mathbf{x}}^{(0)}$, где $\mathbf{x}^H \in S_1^c$, $\tilde{\mathbf{x}}^{(0)} \in S_0^c$, то

$$\begin{aligned} \mathbf{u}^{(k)} - \mathbf{u} &= \mathbf{y}^{(k)} - k(\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}) - \mathbf{u} = B^{-\frac{1}{2}} [\mathbf{x}^{(k)} - k(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)})] - \mathbf{u} = \\ &= B^{-\frac{1}{2}} [\bar{\mathbf{x}}^{(k)} + \tilde{\mathbf{x}}^{(0)} + k\tau\tilde{\mathbf{g}} - k(\bar{\mathbf{x}}^{(k+1)} + \tilde{\mathbf{x}}^{(0)} + k\tau\tilde{\mathbf{g}} + \tau\tilde{\mathbf{g}} - \bar{\mathbf{x}}^{(k)} - \\ &- \tilde{\mathbf{x}}^{(0)} - k\tau\tilde{\mathbf{g}}) - \mathbf{x}^H - \tilde{\mathbf{x}}^{(0)}] = B^{-\frac{1}{2}} [\bar{\mathbf{x}}^{(k)} - \mathbf{x}^H - k(\bar{\mathbf{x}}^{(k+1)} - \bar{\mathbf{x}}^{(k)})] = \\ &= B^{-\frac{1}{2}} [\bar{\mathbf{z}}^{(k)} - k(\bar{\mathbf{z}}^{(k+1)} - \bar{\mathbf{z}}^{(k)})]. \end{aligned} \quad (\text{IV.65})$$

Из последнего равенства получаем

$$\begin{aligned} \|\mathbf{u}^{(k)} - \mathbf{u}\| &= \|B^{-\frac{1}{2}} [\bar{\mathbf{z}}^{(k)} - k(\bar{\mathbf{z}}^{(k+1)} - \bar{\mathbf{z}}^{(k)})]\| \leq \|B^{-\frac{1}{2}}\| (\|\bar{\mathbf{z}}^{(k)}\| + \\ &+ k(\|\bar{\mathbf{z}}^{(k+1)}\| + \|\bar{\mathbf{z}}^{(k)}\|)) \leq \|B^{-\frac{1}{2}}\| [1 + k(1 + \rho)] \|\bar{\mathbf{z}}^{(k)}\| \leq \\ &\leq \|B^{-\frac{1}{2}}\| [1 + k(1 + \rho)] \rho^k \|\bar{\mathbf{z}}^{(0)}\|, \end{aligned}$$

т. е. $\|\mathbf{u}^{(k)} - \mathbf{u}\| \leq \|B^{-\frac{1}{2}}\| [1 + k(1 + \rho)] \rho^k \|\bar{\mathbf{z}}^{(0)}\|$.

Отсюда, учитывая, что $\rho < 1$, следует

$$\lim_{k \rightarrow \infty} \|\mathbf{u}^{(k)} - \mathbf{u}\| = 0.$$

Если для $\mathbf{y}^{(k+1)}$ и $\mathbf{y}^{(k)}$ итерационной схемы выполняется неравенство (IV.54), то справедлива следующая оценка:

$$\|\mathbf{u}^{(k)} - \mathbf{u}\| \leq c_k \|\mathbf{r}^{(k)} - \mathbf{r}^{(k-1)}\| \leq c_k \varepsilon, \quad (\text{IV.66})$$

где $c_k = \frac{1 + k\tau\gamma_2}{\tau\gamma_1^2 v}$, v — минимальное собственное значение матрицы B .

Покажем это. В (IV.65) установлено, что

$$\mathbf{u}^{(k)} - \mathbf{u} = B^{-\frac{1}{2}} [\bar{\mathbf{z}}^{(k)} - k(\bar{\mathbf{z}}^{(k+1)} - \bar{\mathbf{z}}^{(k)})]. \quad (\text{IV.67})$$

Умножая (IV.61) на $\boldsymbol{\psi}_j$ и суммируя произведения от $j = m+1$ до n , получаем

$$\bar{\mathbf{z}}^{(k+1)} = (E - \tau C) \bar{\mathbf{z}}^{(k)}.$$

Подставляя $\bar{\mathbf{z}}^{(k+1)}$ в (IV.67), можно записать

$$\mathbf{u}^{(k)} - \mathbf{u} = B^{-\frac{1}{2}} (E + k\tau C) \bar{\mathbf{z}}^{(k)}.$$

В то же время можно показать, как это было сделано выше, что

$$B^{-\frac{1}{2}}(\mathbf{r}^{(k)} - \mathbf{r}^{(k+1)}) = C(\bar{\mathbf{z}}^{(k)} - \bar{\mathbf{z}}^{(k+1)}) = \tau C^2 \bar{\mathbf{z}}^{(k)},$$

или

$$\bar{\mathbf{z}}^{(k)} = \frac{1}{\tau} (C^2)^{-1} B^{-\frac{1}{2}} (\mathbf{r}^{(k)} - \mathbf{r}^{(k+1)}),$$

$$\mathbf{u}^{(k)} - \mathbf{u} = \frac{1}{\tau} B^{-\frac{1}{2}} (E + k\tau C) (C^2)^{-1} B^{-\frac{1}{2}} (\mathbf{r}^{(k)} - \mathbf{r}^{(k+1)}),$$

откуда и следует оценка (IV.66). Справедливо неравенство

$$\|\mathbf{u}^{(k)} - \mathbf{u}\| \leq q_k \|\bar{\mathbf{z}}^{(0)}\|,$$

где

$$q_k = \frac{1}{\nu} \rho^k [1 + k(1 + \rho)].$$

Покажем на примере несовместной системы линейных алгебраических уравнений третьего порядка с симметричной и положительно полуопределенной матрицей работу итерационного процесса, записанного в виде (IV.42) в случае, когда $B = E$ и $\mathbf{y}^{(0)} = 0$. Рассматриваемая система обладает ортогональной системой собственных векторов $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, соответствующих собственным значениям $\lambda_1, \lambda_2, \lambda_3$. Пусть $\lambda_1 = 0$ (рис. 10). Нетрудно убедиться, что обобщенное решение такой системы находится на прямой AB , точка C — нормальное псевдорешение. Система несовместна по той причине, что правая часть неортогональна $\mathbf{v}_1$. Проекция $\mathbf{f}$ на $\mathbf{v}_1$ есть $\tilde{\mathbf{f}}$ (рис. 11). При заданном нулевом начальном приближении после первой итерации проекция приближения $\mathbf{y}^{(1)}$ на $\mathbf{v}_1$ равна $\tau\tilde{\mathbf{f}}$, хотя проекция погрешности в пространстве S_1^A , т. е. в плоскости $\mathbf{v}_2, \mathbf{v}_3$, уменьшается. После второй итерации проекция $\mathbf{y}^{(2)}$ на $\mathbf{v}_1$ будет равна $2\tau\tilde{\mathbf{f}}$, хотя проекция погрешности $\mathbf{y}^{(2)}$ на плоскость $\mathbf{v}_2, \mathbf{v}_3$ снова уменьшается и т. д.

Таким образом, построенный итерационный процесс в пространстве S^A расходится, хотя в подпространстве S_1^A он дает сходимость приближений к нормальному псевдорешению. С помощью формулы (IV.50) можно уничтожить проекцию $\mathbf{v}_1$ после того, как достигнута сходимость проекций решения в подпространстве S_1^A .

При решении прикладных задач на ЭВМ возникают системы линейных алгебраических уравнений с матрицами произвольного вида (прямоугольными, квадратными вырожденными, квадратными невырожденными и т. д.).

Численные методы вычисления единственного классического решения в зависимости от свойств матрицы и погрешности в задании исходных данных были рассмотрены в первой и второй главах. В ряде случаев при решении систем с прямоугольными и квадратными вырож-

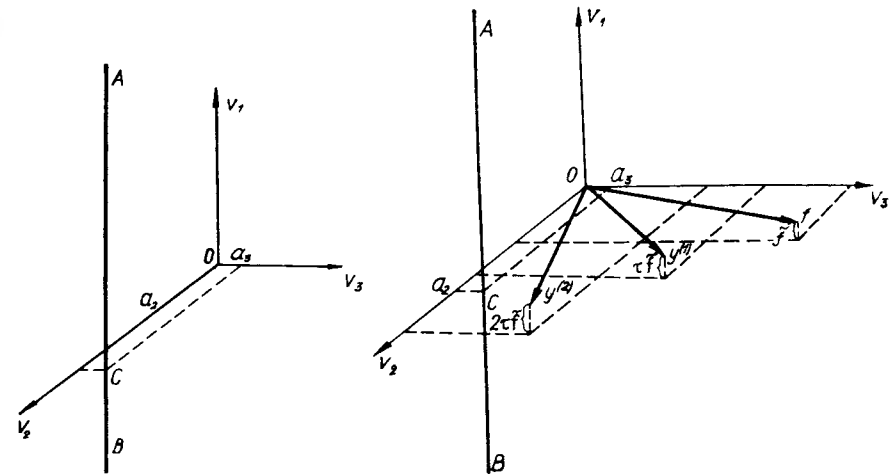


Рис. 10

Рис. 11

денными матрицами приходится вместо классического разыскивать обобщенное решение задачи или псевдообратную матрицу.

Известно, что псевдообратная матрица $A^\#$ удовлетворяет четырем уравнениям Пенроуза

$$AA^\#A = A, (AA^\#)^* = AA^\#, A^\#AA^\# = A^\#, (A^\#A)^* = A^\#A.$$

Для системы линейных алгебраических уравнений

$$A\mathbf{x} = \mathbf{b}$$

решение

$$\mathbf{x} = A^\#\mathbf{b}$$

является нормальным обобщенным решением по методу наименьших квадратов.

Ряд методов обобщенного обращения матриц основано на представлении матрицы A с размерами $m \times n$ ранга r (A) в виде

$$A = BC,$$

где B — прямоугольная матрица с размерами $m \times r$, C — матрица с размерами $r \times n$. Из этого представления следует, что

$$A^\# = C^\#B^\#, C^\# = C^*(CC^*)^{-1}, B^\# = (B^*B)^{-1}B^*.$$

Как отмечалось выше, матрицу A можно представить в виде сингулярного разложения

$$A = UDV^T,$$

где $D = \text{diag } \sigma_i$, σ_i — сингулярные числа матрицы A , V и U — матрицы, составленные из ортонормированных собственных векторов матриц $A^T A$ и AA^T соответственно, тогда

$$A^\# = VD^\#U^T.$$

Вычисление псевдообратной матрицы с помощью сингулярного разложения требует $(7 \div 8)n^3$ мультипликативных операций. Псевдообрат-

ные матрицы могут быть получены и с помощью итерационных процессов [125, 127].

В зависимости от постановки задачи, вида и свойств матрицы, технических и математических характеристик имеющейся ЭВМ применяют те или иные способы получения одного из обобщенных решений задачи. Если матрица системы и ее правая часть целиком помещаются в оперативную память ЭВМ, то для получения псевдообратных матриц, построения нормального псевдорешения, вычисления нетривиального решения однородной системы в случае, когда $r(A) \leq \min(m, n)$, целесообразно использовать метод, основанный на сингулярном разложении матриц. А когда перемножение матриц A^T и A в пределах длины машинного слова ведет к незначительному накоплению погрешностей для построения нормального псевдорешения, можно использовать метод регуляризации А. Н. Тихонова, который работает примерно в три раза быстрее, чем метод, основанный на сингулярном разложении матриц.

Если матрица системы линейных алгебраических уравнений симметрична, положительно полуопределена и известна оценка снизу первого отличного от нуля собственного значения матрицы системы, то применение сдвига спектра матрицы и последующее использование для решения системы прямого метода (LL^T - или LDL^T -разложения) позволяет получить приближение к одному из обобщенных решений за меньшее число машинных операций по сравнению с методом сингулярного разложения матрицы или методом регуляризации А. Н. Тихонова.

Если симметричная и положительно полуопределенная матрица системы и ее правая часть не помещаются в оперативную память ЭВМ, а матрица A — порождающаяся, причем программа порождения, правая часть b и приближение к решению помещаются в оперативную память ЭВМ, и кроме того известны оценки снизу для первого, отличного от нулевого, и сверху для максимального собственных значений матрицы, то в этом случае можно использовать для получения приближения к обобщенному решению итерационные методы [65]. Простейшим из этих методов является метод, реализуемый канонической формой (IV.42) в случае, когда $B = E$. При этом можно использовать как постоянный, так и переменный набор параметров, ускоряющих сходимость проекций решения на подпространство, натянутое на собственные векторы, отвечающие ненулевым собственным значениям.

Если для системы линейных алгебраических уравнений с симметричной и положительно полуопределенной матрицей можно построить перестановочные с A и легко обратимые матрицы B , то целесообразно использовать эти матрицы для ускорения сходимости итераций.

Если позволяет оперативная память машины, то вместо одношаговых итерационных процессов целесообразно применять двухшаговые итерационные процессы, которые в канонической форме А. А. Самарского можно записать в следующем виде:

$$B \left[\frac{y^{(k+1)} - y^{(k-1)}}{2\tau} + \kappa (y^{(k+1)} - 2y^{(k)} + y^{(k-1)}) \right] + Ay^{(k)} = f, \quad y^{(0)} = y_0, \\ y^{(1)} = y_1,$$

где B — положительно определенная перестановочная с A матрица, τ и κ — параметры, обеспечивающие сходимость итерационного процесса

$$\tau = \frac{1}{|\gamma_1 \gamma_2|}, \quad \kappa = \frac{\gamma_1 + \gamma_2}{4},$$

γ_1 — оценка снизу первого, отличного от нулевого, γ_2 — оценка сверху максимального собственных значений матрицы $B^{\frac{1}{2}}AB^{-\frac{1}{2}}$, вычисляемые неявно.

Итерационным методам получения обобщенного решения для систем линейных алгебраических уравнений с несимметричными матрицами посвящены работы [127, 128].

Отметим, что исследования в области методов построения обобщенных решений систем линейных алгебраических уравнений сосредоточиваются вокруг создания различных методов получения обобщенных решений и псевдообращения матриц, построения итерационных методов получения приближений к обобщенным решениям, исследования вопросов машинной реализации методов построения различных обобщенных решений и псевдообращения матриц.

МЕТОДЫ РЕШЕНИЯ НЕЛИНЕЙНЫХ
АЛГЕБРАИЧЕСКИХ И ТРАНСЦЕНДЕНТНЫХ
УРАВНЕНИЙ

V.1. ПОСТАНОВКА ЗАДАЧИ, ОПРЕДЕЛЕНИЯ,
НЕКОТОРЫЕ ОБЩИЕ СВЕДЕНИЯ

1. Задача об отражении звуковой волны на границе раздела двух сред [90]. Пусть звуковая волна

$$y_1 = a_1 \cos \left(\omega t - \frac{2\pi x}{\lambda_1} \right) \quad (V.1)$$

падает под прямым углом на границу раздела сред AA' (см. рис. 12), а источник волны S находится в среде I . В формуле (V.1) a_1 — амплитуда волны, ω — частота, t — время, x — координата вдоль оси падения волны, λ_1 — длина волны.

Точку на границе раздела двух сред выберем в качестве начала координат. От границы раздела сред AA' идет отраженная волна

$$y_2 = a_2 \cos \left(\omega t + \frac{2\pi x}{\lambda_1} + \varphi_2 \right). \quad (V.2)$$

Но через границу раздела проходит некоторая волна

$$y_3 = a_3 \cos \left(\omega t + \frac{2\pi x}{\lambda_2} - \varphi_3 \right), \quad (V.3)$$

которая распространяется во второй среде. На границе раздела сред в любой момент времени t выполняются следующие очевидные законы: непрерывности смещений, т. е.

$$y_1 + y_2 = y_3, \quad (V.4)$$

равенства давлений

$$p_1 + p_2 = p_3. \quad (V.5)$$

Пусть известны амплитуда падающей волны a_1 , длины падающей λ_1 , и отраженной λ_2 волн, а также некоторые постоянные сред I и II (см. рис. 12) κ_1 и κ_2 . Требуется определить амплитуды отраженной и проходящей волн a_2 и a_3 и их фазы φ_2 и φ_3 . Для решения этой задачи необходимо написать уравнения относительно a_2 , a_3 , φ_2 и φ_3 .

Подставляя формулы (V.1) — (V.3) в (V.4), для $x = 0$ получаем

$$a_1 \cos \omega t + a_2 \cos (\omega t + \varphi_2) = a_3 \cos (\omega t - \varphi_3). \quad (V.6)$$

Давление в плоской волне определяют по формуле

$$p = -\kappa \rho_0 \frac{\partial y}{\partial x}, \quad (V.7)$$

где κ — постоянная среды, а ρ_0 — физическая постоянная. Учитывая эту формулу, определяем давление в каждой волне:

$$p_1 = -\kappa_1 \rho_0 a_1 \frac{2\pi}{\lambda_1} \sin \left(\omega t - \frac{2\pi x}{\lambda_1} \right), \quad (V.8)$$

$$p_2 = \kappa_1 \rho_0 a_2 \frac{2\pi}{\lambda_1} \sin \left(\omega t + \frac{2\pi x}{\lambda_1} + \varphi_2 \right),$$

$$p_3 = \kappa_2 \rho_0 a_3 \frac{2\pi}{\lambda_2} \sin \left(\omega t + \frac{2\pi x}{\lambda_2} - \varphi_3 \right).$$

В точке $x = 0$ выполняется равенство (V.5). Подставляя в него (V.8) и деля на $2\pi \rho_0 \neq 0$, можно записать

$$\begin{aligned} \frac{\kappa_1}{\lambda_1} (-a_1 \sin \omega t + a_2 \sin (\omega t + \varphi_2)) = \\ = \frac{\kappa_2}{\lambda_2} a_3 \sin (\omega t - \varphi_3). \end{aligned} \quad (V.9)$$

Учитывая известные равенства

$$\sin (A \pm B) = \sin A \cos B \pm \sin B \cos A,$$

$$\cos (A \pm B) = \cos A \cos B \mp \sin A \sin B,$$

формулы (V.6) и (V.9) можно переписать следующим образом:

$$(a_1 + a_2 \cos \varphi_2 - a_3 \cos \varphi_3) \cos \omega t - (a_2 \sin \varphi_2 + a_3 \sin \varphi_3) \sin \omega t = 0,$$

$$\begin{aligned} \left[\frac{\kappa_1}{\lambda_1} (a_2 \cos \varphi_2 - a_1) - \frac{\kappa_2}{\lambda_2} a_3 \cos \varphi_3 \right] \sin \omega t + \\ + \left(\frac{\kappa_1}{\lambda_1} a_2 \sin \varphi_2 + \frac{\kappa_2}{\lambda_2} a_3 \sin \varphi_3 \right) \cos \omega t = 0. \end{aligned} \quad (V.10)$$

Так как равенство (V.10) выполняется в любой момент времени t , приравнявая коэффициенты при $\sin \omega t$ и $\cos \omega t$, получаем систему нелинейных уравнений относительно a_2 , a_3 , φ_2 , φ_3 :

$$a_2 \cos \varphi_2 - a_3 \cos \varphi_3 + a_1 = 0,$$

$$a_2 \sin \varphi_2 + a_3 \sin \varphi_3 = 0,$$

$$\frac{\kappa_1}{\lambda_1} (a_2 \cos \varphi_2 - a_1) - \frac{\kappa_2}{\lambda_2} a_3 \cos \varphi_3 = 0,$$

$$\frac{\kappa_1}{\lambda_1} a_2 \sin \varphi_2 + \frac{\kappa_2}{\lambda_2} a_3 \sin \varphi_3 = 0.$$

Решив систему уравнений (V.11), можно определить амплитуды a_2 , a_3 и их фазы φ_2 , φ_3 .

2. Системы нелинейных уравнений. Система нелинейных уравнений в общем случае имеет вид

$$f_i(x_1, x_2, \dots, x_n) = 0, \quad i = 1, 2, \dots, n. \quad (V.12)$$

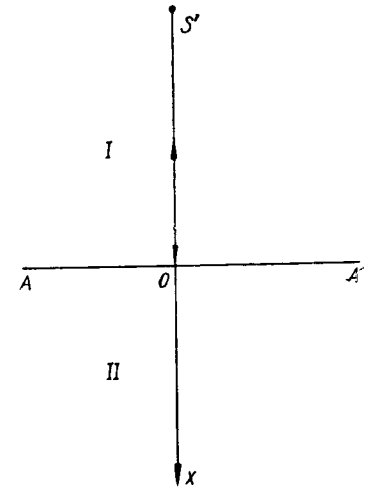


Рис. 12

Операторная форма этой системы следующая:

$$F(x) = 0, \quad (V.13)$$

где $x = (x_1, x_2, \dots, x_n)$, $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$.

Системы нелинейных уравнений могут быть переопределенными, если число уравнений больше числа неизвестных. Возможны случаи, когда число неизвестных в системе (V.13) больше числа уравнений. Но эти системы нелинейных уравнений нами рассматриваться не будут. Впредь будем рассматривать лишь системы, в которых число уравнений равно числу неизвестных. Кроме того, будем находить лишь действительные решения. Такие системы нелинейных уравнений могут не иметь решения, иметь единственное решение или иметь множество решений (конечное либо бесконечное).

Рассмотрим систему [72]

$$x_1^2 - x_2 + \alpha = 0, \quad x_1 + x_2^2 + \alpha = 0, \quad (V.14)$$

где α — числовой параметр. При $\alpha = 1$ не существует решения такой системы в действительных числах, так как нет таких действительных чисел c_1 и c_2 , которые одновременно удовлетворяли бы системе (V.14). Если $\alpha = 1/4$, то решение единственно и определяется по формуле $x_1 = x_2 = 1/2$. При $\alpha = 0$ решений системы (V.14) два: $x_1 = x_2 = 0$, $x_1 = x_2 = 1$. Приняв $\alpha = -1$, получим четыре решения: $x_1 = -1$, $x_2 = 0$, $x_1 = 0$, $x_2 = 1$, $x_1 = x_2 = \frac{(1 \pm \sqrt{5})}{2}$.

Система нелинейных уравнений

$$\frac{1}{2} x_1 \left[\sin \left(\frac{1}{2} \pi x_1 \right) \right] - x_2 = 0, \quad x_2^2 - x_1 + 1 = 0$$

имеет счетное множество решений. А система

$$x_1^2 - |x_2| = 0, \quad x_1^2 - x_2 = 0$$

имеет континуум решений.

Введем определение производных Гато и Фреше, следуя [72]. Отображение $F: D \subset R^n \rightarrow R^m$ называется дифференцируемым по Гато (или G -дифференцируемым) во внутренней точке множества D , если существует такой линейный оператор $A \in L(R^n, R^m)$, что для любого $h \in R^n$

$$\lim_{t \rightarrow 0} \left(\frac{1}{t} \right) \| F(x + th) - Fx - tAh \| = 0. \quad (V.15)$$

Если отображение $F: D \subset R^n \rightarrow R^m$ является G -дифференцируемым в точке $x \in \text{int}(D)$, то тот единственный линейный оператор $A \in L(R^n, R^m)$, для которого выполняется равенство (V.15), называется G -производной отображения F в точке x и обозначается через $F'(x)$.

Отображение $F: D \subset R^n \rightarrow R^m$ называется дифференцируемым по Фреше (или F -дифференцируемым) в точке $x \in \text{int}(D)$, если существует такой оператор $A \in L(R^n, R^m)$, что

$$\lim_{h \rightarrow 0} \left(\frac{1}{\|h\|} \right) \| F(x + h) - Fx - Ah \| = 0. \quad (V.16)$$

Этот линейный оператор A обозначается тоже через $F'(x)$ и называется F -производной отображения F в точке x .

Заметим, что из существования предела (V.16) вытекает существование предела (V.15), но не наоборот. Таким образом, отображение F является G -дифференцируемым в точке x , если оно F -дифференцируемо в этой точке.

Матричное представление производной $F'(x)$ дается матрицей Якоби

$$F'(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}. \quad (V.17)$$

Важно отметить, что из существования матрицы Якоби, т. е. из существования всех частных производных еще не вытекает G -дифференцируемость отображения F . Более того даже существование предела

$$\lim_{t \rightarrow 0} \left(\frac{1}{t} \right) [F(x + th) - Fx]$$

для всех $h \in R^n$ не означает, что отображение F имеет G -производную в точке x .

Теоремы единственности и существования решения нелинейных уравнений можно найти в работах [72, 116].

В дальнейшем при рассмотрении систем нелинейных уравнений будем предполагать, что решение системы (V.13) существует и единственно.

3. Нелинейные уравнения с одним неизвестным. Рассмотрим нелинейное уравнение с одним неизвестным

$$f(x) = 0. \quad (V.18)$$

Будем считать, что $f(x)$ — непрерывная однозначная на отрезке $[a, b]$ функция. Здесь a и b могут принимать любые значения, в том числе $\pm \infty$. Более того будем считать, что x принадлежит некоторой комплексной области D , т. е. $f(x)$ может быть функцией комплексной переменной.

Всякое число x^* из области определения функции такое, что

$$f(x^*) = 0,$$

называется корнем уравнения (V.18) или нулем функции $f(x)$.

Пусть $f(x)$ — гладкая функция. Число x^* называется корнем k -й кратности, если при $x = x^*$ вместе с функцией $f(x)$ обращаются в нуль ее производные до $(k-1)$ -го порядка включительно, т. е.

$$f(x^*) = f'(x^*) = \dots = f^{(k-1)}(x^*) = 0, \quad f^{(k)}(x^*) \neq 0.$$

Однократный корень называется простым. Уравнение (V.18) называется алгебраическим, если $f(x)$ — многочлен n -й степени от одной

или нескольких переменных. Алгебраическим уравнением с одним неизвестным называют уравнение вида

$$P_n(x) = x^n + a_{n-1}x^{n-1} + \dots + a_0 = 0, \quad (V.19)$$

где n — целое неотрицательное число, называемое степенью уравнения, $a_0, a_1, \dots, a_{n-1}$ — коэффициенты уравнения, x — неизвестное, которое необходимо определить.

Известно, что алгебраическое уравнение (V.19) степени n с комплексными коэффициентами имеет ровно n корней, если каждый k -кратный корень считать столько раз, какова его кратность. Для алгебраических уравнений существует ряд утверждений, позволяющих определить число корней и их границы. Приведем без доказательства некоторые из них.

Отметим, что задачу по определению корней нелинейного алгебраического уравнения с комплексными коэффициентами можно свести к решению нелинейного уравнения с действительными коэффициентами. Для этого необходимо умножить исходное уравнение (V.19) на многочлен, коэффициенты которого комплексно-сопряжены с коэффициентами исходного уравнения

$$P_{2n}(x) = P_n(x) \bar{P}_n(x).$$

Степень полученного полинома будет $2n$, и его корни будут содержать корни исходного уравнения (V.19).

Остаток от деления полинома $P_n(x)$ на одночлен $(x - a)$ равен значению этого полинома при $x = a$, т. е. $P_n(a)$. Если известен корень уравнения (V.19) $x = x^*$, то степень уравнения можно понизить, разделив его на $x - x^*$, т. е.

$$P_{n-1}(x) = \frac{P_n(x)}{x - x^*},$$

при этом остальные корни исходного уравнения будут равны корням полученного уравнения.

Для нелинейных алгебраических уравнений справедливы следующие известные утверждения.

Пусть функция $f(x)$ непрерывна на отрезке $[a, b]$. Если $f(a)f(b) < 0$, то между a и b имеется нечетное число корней. Если $f(a)f(b) > 0$, то между a и b имеется четное число корней или их вовсе нет.

Если функция $f(x)$ дифференцируема на отрезке $[a, b]$, $f(a)f(b) < 0$ и производная $f'(x)$ не меняет своего знака на рассматриваемом отрезке, то между a и b имеется один корень уравнения (V.18). Более подробно свойства корней нелинейных алгебраических уравнений изложены, например, в работе [6-7, т. 2].

Если функция $f(x)$ не является алгебраической, то уравнение (V.18) называется трансцендентным. Примером такого уравнения является

$$\operatorname{tg} z - \frac{1}{z} = 0.$$

В некоторых случаях решение трансцендентного уравнения можно свести к решению нелинейного алгебраического уравнения. В общем случае трансцендентные уравнения решаются только численными методами.

V.2. ЧИСЛЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ С ОДНИМ НЕИЗВЕСТНЫМ

1. Нелинейные алгебраические уравнения, описывающие физические модели. Решение прикладных задач, как правило, начинается с создания приемлемых физических и математических моделей. При описании физических моделей редко возникают нелинейные уравнения

$$\tilde{f}(x) = 0, \quad a \leq x \leq b \quad (V.20)$$

с точными исходными данными. Наиболее типичной является постановка приближенного уравнения

$$f(x) = 0, \quad a \leq x \leq b \quad (V.21)$$

с указанием погрешности в исходных данных:

$$|\tilde{f}(x) - f(x)| < \varepsilon, \quad a \leq x \leq b. \quad (V.22)$$

Рассмотрим некоторые примеры трудностей, возникающих при рассмотрении задачи (V.21), (V.22). Пусть имеется уравнение

$$e^x = \varepsilon \quad (V.23)$$

с решением $x = \ln \varepsilon$, где ε — малое число. Ошибка в исходных данных может привести к уравнению

$$e^x = 0,$$

не имеющему конечного решения, так как не существует конечного значения x , для которого выполняется равенство (V.23). Нетрудно убедиться, что уравнение

$$x^2 - 2x + 1 = 0 \quad (V.24)$$

имеет корни $x_1 = 1, x_2 = 1$, а уравнение

$$x^2 - 2x + 1,01 = 0$$

обладает корнями $x_1 = 1 + 0,1i, x_2 = 1 - 0,1i$. В этом примере погрешность в исходных данных вывела корни уравнения (V.24) из рассматриваемой области действительных чисел и перевела их в область комплексных чисел.

Уравнение

$$x^{1/3} - 2x^{1/6} + 1 = 0$$

имеет действительный корень $x^* = 1$, а уравнение

$$x^{1/3} - 2x^{1/6} + 0,99 = 0, \quad (V.25)$$

отличающееся от предыдущего лишь погрешностью в свободном члене $\varepsilon = 10^{-2}$, имеет два действительных корня $x_1 = 1,771\,561, x_2 = 0,531\,441$.

Решение уравнения (V.20) называется устойчивым, если малым изменениям в исходных данных соответствуют малые изменения в решении.

Иными словами, если $|\tilde{f}(x) - f(x)| < \varepsilon$ при всех $x \in [a, b]$, то и

$$|\tilde{x}^* - x^*| \leq \delta,$$

где $\tilde{x}^*$ — точное решение уравнения (V.20), x^* — точное решение задачи (V.21), $\varepsilon, \delta = \delta(\varepsilon)$ — как угодно малые числа.

Рассмотрим условия, при которых решение задачи (V.21), (V.22) может быть принято за решение уравнения (V.20). Для этого оценим разность решений уравнений (V.20), (V.21) через погрешность в исходных данных и функции уравнения (V.21). Нетрудно убедиться, что справедливо равенство

$$f(\tilde{x}^*) = f(x^*) - f'(\theta)(x^* - \tilde{x}^*), \quad (V.26)$$

где $\theta \in [\tilde{x}^*, x^*]$.

Из (V.26) можно записать

$$x^* - \tilde{x}^* = -\frac{f(\tilde{x}^*)}{f'(\theta)} \text{ при } f'(\theta) \neq 0. \quad (V.27)$$

Учитывая уравнение (V.20) и обозначение

$$\tilde{f}(\tilde{x}^*) - f(\tilde{x}^*) = \delta f,$$

из (V.27) получаем

$$x^* - \tilde{x}^* = \frac{\delta f}{f'(\theta)}. \quad (V.28)$$

Учитывая равенство (V.28), получаем оценку

$$|x^* - \tilde{x}^*| \leq \frac{|\delta f|}{\min_{a \leq x \leq b} |f'(x)|}. \quad (V.29)$$

Назовем величину

$$M = \frac{1}{\min_{a \leq x \leq b} |f'(x)|} \quad (V.30)$$

локальным числом обусловленности функции $f(x)$. При этом величина, которая будет характеризовать обусловленность задачи (V.21)

$$m = \frac{|\delta f|}{\min_{a \leq x \leq b} |f'(x)|} = M |\delta f|. \quad (V.31)$$

Чем больше значение m на отрезке, где находится корень, тем хуже обусловленность задачи.

Так, в нелинейном уравнении (V.25) $\delta f = 0,01$, $f'(x) = -\frac{1}{3}(x^{-\frac{4}{6}} - x^{-\frac{4}{3}})$ и $\min f'(x) = 0$ при $x = 1$. Следовательно, эта функция плохо обусловлена, а из оценки (V.29) вытекает

$$|x^* - \tilde{x}^*| \leq \infty.$$

Хотя эта оценка носит мажорантный характер, она, естественно, должна настораживать и свидетельствовать о том, что необходим дополнительный анализ получаемого математического решения задачи.

Из рассмотрения чисел обусловленности (V.30) видно, что каждый корень обладает своим числом обусловленности, зная которое мож-

но оценить наследственную погрешность получаемого математического решения.

2. Отделение корней. Задачу по численному нахождению действительных или комплексных корней нелинейных уравнений можно разделить на две части: отделение корней, т. е. нахождение достаточно малых окрестностей корня, в которых он один (или определения начального приближения к разыскиваемому корню), и вычисление корней в заданных интервалах с необходимой точностью.

Рассмотрим некоторые методы отделения корней. Одним из простых методов отделения действительных корней является графический. При этом строят график функции

$$y = f(x) \quad (V.32)$$

и определяют точки пересечения графика с осью x . Абсциссы точек пересечения графика функций принимают как приближение к действительным корням уравнения. Так, функция уравнения

$$x^3 + 3x^2 - 1 = 0 \quad (V.33)$$

изображена на рис. 13.

Некоторые уравнения удобно записать в виде

$$f_1(x) = f_2(x).$$

Абсциссы точек пересечения графиков

$$y_1 = f_1(x), \quad y_2 = f_2(x)$$

принимают в качестве приближенных значений корней.

Так, для уравнения (V.33) можно построить графики функций (рис. 14)

$$y = -x^3, \quad y = 3x^2 - 1.$$

Из рисунка 14 видно, что точки $x_1 = -2,9$; $x_2 = -0,65$; $x_3 = 0,5$ являются приближенными значениями корней. Графический способ удобен для ЭВМ с дисплеем. В этом случае графики можно выводить на экран дисплея. Графики функций можно также выводить на графический построитель. Возможен также табличный способ отделения корней. Он используется, например, при нахождении корней с помощью микрокалькуляторов.

В табличном способе отрезок $[a, b]$ разбивают точками x_i , $i = 1, 2, \dots, n$, на ряд отрезков. Затем вычисляют значения функций $f(x)$ в этих точках. Если $f(x_i)$ и $f(x_{i+1})$ имеют разные знаки, то на отрезке $[x_i, x_{i+1}]$ имеется, по крайней мере, один корень. Если же точки x_i и x_{i+1} недостаточно близки и там должны существовать еще корни, то отрезок $[x_i, x_{i+1}]$ может быть разбит на несколько частей. Таким образом, можно найти достаточно малый отрезок, на концах которого функция принимает разные знаки. Перед началом вычислений определяют некоторые свойства функции (например, поведение $f(x)$ при $x \rightarrow \pm\infty$), значения x , при которых легко вычисляются $f(x)$ (например, в точках $x = 0, \pm 1, \dots$), значения x , при которых $f(x) = 0, \pm\infty$, количество корней нелинейного алгебраического уравнения и т. д.

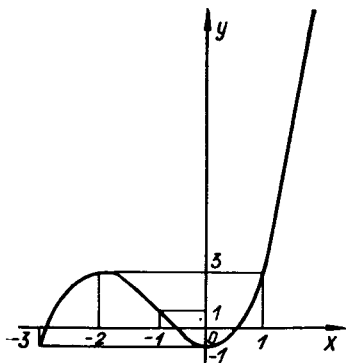


Рис. 13

Иногда функцию $f(x)$ можно приблизить некоторой функцией $\varphi(x)$, корни которой примерно равны исходной. Хорошее приближение к исходным корням нелинейного уравнения иногда можно получить, рассматривая физику процесса, описываемого этим уравнением.

3. Теоремы о неподвижной точке [82]. В связи с большим разнообразием приложений рассмотрим уравнение вида $x = Tx$ и обсудим разновидности теорем существования для таких уравнений — так называемые теоремы о неподвижной точке. Вначале дадим некоторые определения.

Пусть $T: X \rightarrow X$ — некоторое отображение. Тогда $x \in X$, для которой $Tx = x$, называется неподвижной точкой отображения T .

Пусть $\langle S, \rho \rangle$ — метрическое пространство. Отображение $T: S \rightarrow S$, для которого $\rho(Tx, Ty) \leq \alpha \rho(x, y)$, называется сжимающим. Если существует $\alpha < 1$, при котором

$$\rho(Tx, Ty) \leq \alpha \rho(x, y), \quad (V.34)$$

то T называется строго сжимающим.

Строго сжимающее отображение полного метрического пространства имеет единственную неподвижную точку (принцип сжимающих отображений). Покажем это.

Прежде всего докажем единственность. Если $Tx = x$, $Ty = y$, то $\rho(Tx, Ty) = \rho(x, y) \leq \alpha \rho(x, y)$, поскольку $\alpha < 1$ и $\rho(x, y) \geq 0$, заключаем, что $\rho(x, y) = 0$, т. е. $x = y$. Для доказательства существования заметим сначала, что T автоматически непрерывно, ибо если $\rho(x, y) < \alpha \varepsilon$, то $\rho(Tx, Ty) < \varepsilon$. Далее, пусть значение x_0 произвольно и $T^n x_0 = x_n$. Покажем, что $\{x_n\}$ — последовательность Коши. Имеем

$$\begin{aligned} \rho(x_n, x_{n-1}) &= \rho(Tx_{n-1}, Tx_{n-2}) \leq \alpha \rho(x_{n-1}, x_{n-2}) \leq \\ &\leq \alpha^2 \rho(x_{n-2}, x_{n-3}) \leq \dots \leq \alpha^{n-1} \rho(x_1, x_0). \end{aligned}$$

Таким образом, если $n > m$, то при $m \rightarrow \infty$

$$\rho(x_n, x_m) \leq \sum_{j=m+1}^n \rho(x_j, x_{j-1}) \leq \alpha^m (1 - \alpha)^{-1} \rho(x_0, x_1) \rightarrow 0.$$

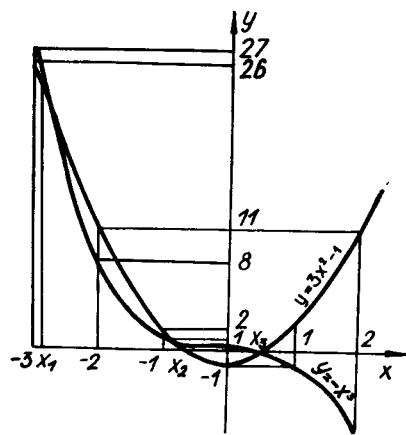


Рис. 14

Итак, $\{x_n\}$ — последовательность Коши, значит, $x_n \rightarrow x$ для некоторого x . Поскольку T непрерывно, $Tx = \lim Tx_n = \lim x_{n+1} = x$, что и доказывает теорему.

Приведем без доказательства теорему, обобщающую известную теорему Брауэра о неподвижной точке.

Пусть C — непустое компактное выпуклое подмножество локально выпуклого пространства X . Пусть $T: C \rightarrow C$ — непрерывное отображение. Тогда T обладает неподвижной точкой.

4. Принцип построения одношаговых итерационных процессов. После того как корни отделены или для них получены хорошие начальные приближения, появляется возможность вычислить их с заданной точностью. Для нахождения изолированного на $[a, b]$ корня уравнения (V.21) можно использовать одношаговые итерационные процессы вида

$$x_{k+1} = \varphi(x_k), \quad k = 0, 1, \dots, \quad (V.35)$$

где x_0 — заданное начальное приближение. Если функция $\varphi(x)$ дифференцируема, то величину α в неравенстве (V.34) можно оценить следующим образом:

$$\alpha \leq \max_{x \in D} |\varphi'(x)|. \quad (V.36)$$

Выбор функции φ в итерационном процессе (V.35) имеет большое значение. Он может быть осуществлен не единственным образом. Далеко не всякий выбор $\varphi(x)$ ведет к желаемой цели.

Пусть, например, требуется найти итерационным методом корни уравнения

$$x^2 - 1 = 0 \quad (V.37)$$

на отрезке $[0, 1; 1, 1]$. Если итерационную схему записать в виде

$$x_{k+1} = x_k^2 - 1 + x_k,$$

т. е. взять $\varphi(x) = x^2 - 1 + x$, то видно, что

$$\varphi'(x) = 2x + 1, \quad 1, 2 = \min_x |\varphi'(x)| \leq \alpha \leq \max_x |\varphi'(x)| = 3, 2$$

Так как коэффициент α больше единицы, итерационный процесс с таким выбором $\varphi(x)$ может и не сходиться. Если же для решения уравнения (V.37) итерационный процесс вести по формуле

$$x_{k+1} = x_k - \frac{1}{3}((x_k)^2 - 1),$$

т. е. принять $\varphi(x) = x - \frac{1}{3}(x^2 - 1)$, то $\varphi'(x) = 1 - \frac{2}{3}x$ и $\max_x |\varphi'(x)| = 1/15$. В этом случае построенный итерационный процесс будет сходящимся.

Но выбор $\varphi(x)$ должен обеспечить не только сходимость итерационного процесса, но и его эффективность. При этом критерием эффективности могут служить минимальное число арифметических операций, необходимых для получения решения задачи, минимальное время ре-

шения задачи на ЭВМ и т. д. Чаще всего в качестве критерия эффективности итерационных методов решения нелинейных уравнений выбирают порядок сходимости итерационного процесса. Пусть для некоторого итерационного процесса справедлива оценка

$$|x_{k+1} - x^*| \leq c |x_k - x^*|^\alpha, \quad (V.38)$$

где $c = c(q)$ — некоторая функция, ограниченная сверху, $q = |x_k - x^*|$, α — порядок (степень, показатель) сходимости итерационного процесса. Из (V.38) следует, что чем больше α , тем быстрее будет сходиться итерационный процесс, если значения x_k достаточно близки к x^* . Если $\alpha = 1$, то считают, что итерационный процесс сходится линейно, а если $\alpha = 2$, то — квадратично и т. д. Предположим, что $c = c(q)$ стремится к нулю при $k \rightarrow \infty$ и $\alpha = 1$. В этом случае считают, что итерационный процесс имеет сверхлинейную сходимость. В некоторых методах $\alpha = \alpha(k)$. Тогда величину

$$\alpha^* = \lim_{k \rightarrow \infty} \alpha(k)$$

называют асимптотической скоростью сходимости. Наша задача состоит в том, чтобы по заданному уравнению (V.21) построить такую функцию $\varphi(x)$, чтобы итерационный процесс не только сходил, но и имел такие параметры α и $c = c(q)$, которые обеспечивали бы высокую скорость сходимости процесса.

Рассмотрим теперь на конкретных примерах, как осуществляется построение $\varphi(x)$ и проводится оценка скорости сходимости итерационных процессов.

Метод простой итерации, рассмотренный для линейных систем в п. 2 параграфа II.2, реализован для уравнения (V.21) следующим образом:

$$x_{k+1} = \varphi(x_k), \text{ где } \varphi(x_k) = x_k - f(x_k). \quad (V.39)$$

Определим условия сходимости метода и оценим скорость его сходимости. Пусть x^* — точное решение уравнения (V.21). Тогда из (V.39) можно записать

$$x_{k+1} - x^* = x_k - x^* - f(x_k).$$

Если $f(x)$ — непрерывно дифференцируемая функция, то последнее равенство можно переписать следующим образом:

$$x_{k+1} - x^* = x_k - x^* - f(x^*) - f'(\theta)(x_k - x^*),$$

где $\theta \in [x_k, x^*]$. Отсюда получим оценку

$$|x_{k+1} - x^*| \leq M |x_k - x^*|,$$

в которой число M таково, что

$$|f'(x)| = |1 - f'(x)| \leq M$$

для всех x , принадлежащих отрезку, на котором ищут корень. Нетрудно видеть, что метод простой итерации будет сходиться, если $M < 1$, причем будет наблюдаться линейная сходимость. Покажем, как построить функцию $\varphi(x)$ так, чтобы повысить порядок сходимости итера-

ционного процесса. Рассмотрим сначала случай, когда корень x^* является простым и уравнение не имеет близких к нему корней. Пусть x_k есть k -е приближение к этому корню. Разложим $f(x)$ из уравнения (V.21) по степеням $x - x_k$:

$$f(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2 + \dots$$

и вместо (V.21) будем рассматривать линейное уравнение

$$f(x_k) + f'(x_k)(x - x_k) = 0,$$

учитывающее лишь два первых члена разложения функции $f(x)$ в ряд Тейлора около точки x_k . Разрешая последнее равенство относительно x , получаем формулу

$$x = x_k - \frac{f(x_k)}{f'(x_k)}, \quad (V.40)$$

определяющую итерационный процесс, известный как метод Ньютона

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \quad (V.41)$$

Предполагая соответствующую гладкость функции $f(x)$, из формулы (V.41) можно записать

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* - \frac{f(x_k) - f(x^*)}{f'(x_k)} = \\ &= x_k - x^* - \frac{f'(x_k)(x_k - x^*) + f''(\theta)(x_k - x^*)^2}{f'(x_k)} = \frac{f''(\theta)}{f'(x_k)} (x_k - x^*)^2, \end{aligned}$$

где $\theta \in [x_k, x^*]$. Из последнего равенства следует оценка

$$|x_{k+1} - x^*| \leq N |x_k - x^*|^2,$$

в которой N удовлетворяет неравенству

$$\max_x \left| \frac{f''(x)}{f'(x)} \right| \leq N.$$

Причем x_k принадлежит отрезку, на котором ищут корень уравнения. Метод Ньютона будет сходиться, если $(x_0 - x^*) N < 1$, причем в этом случае будет наблюдаться квадратичная сходимость.

Таким образом, на каждой итерации разность $(x_k - x^*)$ возводится в квадрат. При $|x_k - x^*| \rightarrow 0$ число правильных десятичных или двоичных цифр примерно удваивается на каждой итерации. Если N порядка единицы, то при выборе начального приближения x_0 таким, что $|x_0 - x^*| < \frac{1}{2}$, имеем

$$|x_1 - x^*| < \frac{1}{2^2}, |x_2 - x^*| < \frac{1}{2^4}, \dots, |x_6 - x^*| < \frac{1}{2^{64}}.$$

Значит, нужно около шести итераций, чтобы сократить ошибку от 0,5 до наименьшего значения, возможного для режима вычислений с плавающей запятой на удвоенной разрядности ЭВМ ЕС.

В том случае, когда корень x^* является двукратным или имеются два очень близких корня x^* и x^{**} , формулу (V.41) использовать нельзя.

В этом случае следует искать более точные значения корней уравнения

$$f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2 = 0. \quad (V.42)$$

При этом наблюдается линейная сходимость метода. Для уточнения корня третьей кратности или трех очень близких корней составляется уравнение третьей степени. Аналогично поступаем с корнями четвертой, пятой кратностей и т. д.

Точность вычисленного корня зависит от критерия окончания итерационного процесса. Его следует выбирать в соответствии с точностью задания исходных данных. Если для метода простой итерации выполнены достаточные условия сходимости к единственному решению нелинейного уравнения, то выполнение на k -й итерации условия

$$\frac{|x_{k+1} - x_k|}{|x_k|} < \inf_x |f'(x)| \frac{\varepsilon}{1 + \varepsilon} \quad (V.43)$$

гарантирует получение решения в ходе итерационного процесса с заданной относительной погрешностью ε , т. е.

$$|x^* - x_k| \leq \varepsilon |x^*|, \quad (V.44)$$

где x^* — точное решение нелинейного уравнения, ε — требуемая точность решения [6-7, т. 2]. Докажем это.

Используя формулу Лагранжа, можно записать

$$x_{k+1} - x_k = f'(\xi)(x^* - x_k), \quad \xi \in [x^*, x_k],$$

откуда

$$|x_{k+1} - x_k| = |f'(\xi)| |x^* - x_k|.$$

Следовательно,

$$|x_{k+1} - x_k| \geq \inf_x |f'(x)| |x^* - x_k|.$$

Из (V.43) следует

$$\inf_x |f'(x)| \geq \frac{|x_{k+1} - x_k|}{|x_k|} \frac{1 + \varepsilon}{\varepsilon}.$$

Из двух последних неравенств можно получить

$$\varepsilon |x_k| \geq (1 + \varepsilon) |x^* - x_k| = |x^* - x_k| + \varepsilon |x^* - x_k|.$$

Так как $|x_k| \leq |x_k - x^*| + |x^*|$, то

$$\varepsilon (|x_k - x^*| + |x^*|) \geq |x^* - x_k| + \varepsilon |x^* - x_k|,$$

откуда следует (V.44), что и требовалось доказать.

Для метода Ньютона условие

$$\frac{|x_{k+1} - x_k|}{|x_k|} \leq (1 - q) \frac{\varepsilon}{1 + \varepsilon}, \quad (V.45)$$

где

$$\left| 1 - \frac{(f'(x_k))^2 - f(x_k)f''(x_k)}{(f'(x_k))^2} \right| < q < 1, \quad |f'(x)| < q < 1$$

гарантирует выполнение неравенства (V.44).

Так как сходимость метода Ньютона имеет лишь локальный характер, т. е. когда начальное приближение находится в окрестности корня, то для получения этого начального приближения можно использовать какой-либо глобально сходящийся метод, например метод половинного деления. Он характеризуется минимальными требованиями, предъявляемыми к информации о функции $f(x)$. Его можно применить, когда функция $f(x)$ непрерывна на $[a, b]$ и $f(a)f(b) < 0$. При заданной точности ε этот метод может быть реализован следующим образом. Отрезок $[a, b]$ делят пополам и, если $f\left(\frac{a+b}{2}\right) \neq 0$, то

выбирают ту из половин $\left[a, \frac{a+b}{2}\right]$, $\left[\frac{a+b}{2}, b\right]$, на концах которой функция $f(x)$ принимает значения разных знаков. Описанные действия повторяют до тех пор, пока не получают такой малый интервал (α, β) , на котором содержится один корень уравнения.

Учитывая систему неравенств

$$\begin{aligned} |x_0 - x^*| &= \left| \frac{b_0 - a_0}{2} - x^* \right| \leq \frac{1}{2} (|b_0 - x^*| + |a_0 - x^*|) \leq \\ &\leq |b_0 - a_0| = b - a, \\ |x_1 - x^*| &\leq |b_1 - a_1| = \frac{b - a}{2}, \\ &\dots \dots \dots \\ |x_k - x^*| &\leq |b_k - a_k| = \frac{b - a}{2^k}, \end{aligned}$$

при $k \rightarrow \infty$, $x_k \rightarrow x$ и условие окончания итераций

$$|\beta - \alpha| \leq \varepsilon,$$

можно записать в виде

$$\beta - \alpha = \frac{b - a}{2^k} \approx \varepsilon.$$

Отсюда следует приближенная оценка числа итераций

$$S = \frac{\ln \frac{b - a}{\varepsilon}}{\ln 2}.$$

Таким образом, теоретически интервал (α, β) может быть сужен до нуля, но его надо сужать настолько, насколько позволяет арифметика ЭВМ в режиме с плавающей запятой. Ясно, что на каждой итерации алгоритма приобретается один бит точности. При критерии $\beta - \alpha \leq 2^{-t}(b - a)$ нужно t итераций. На машинах ЕС ЭВМ в арифметике с удвоенной разрядностью необходимо около 56 итераций, чтобы сократить исходный интервал (1, 16) до двух последовательных чисел с плавающей запятой.

Если каждое вычисление $f(x)$ несложное, то этот метод может быть использован для самостоятельного решения уравнения (V.21).

Отметим, что для реализации метода Ньютона по формуле (V.41) необходимо существование первой производной функции $f(x)$, что

ограничивает применение этого метода. При решении уравнений, для которых вычисление $f'(x)$ затруднено, целесообразно использовать метод секущих (см. п. 6 настоящего параграфа).

5. Определение всех действительных корней на отрезке. В некоторых задачах возникает необходимость найти все действительные корни уравнения (V.21) на отрезке $[a, b]$. Для этого можно использовать одношаговый итерационный метод, предложенный в [83]. Для сходимости этого метода достаточно, чтобы функция $f(x)$ была непрерывна, а производная ограничена. Этот метод сходится и в том случае, когда производная $f'(x)$ имеет разрыв первого рода. Вычислительная схема этого метода может быть реализована следующим образом. До начала итерационного процесса задают критерии окончания счета одного корня

$\frac{|x^* - x_k|}{|x^*|} \leq \varepsilon$, где ε — достаточно малое число, большее машинного нуля, и выбирают некоторое число M , удовлетворяющее неравенству

$$M \geq \max_{a \leq x \leq b} |f'(x)|. \quad (\text{V.46})$$

Заметим, что завышение значения M не нарушает сходимости метода, а только замедляет ее. В качестве начального приближения к решению x_0 берут левый конец отрезка $[a, b]$, т. е. полагают $x_0 = a$.

Для каждого $k = 0, 1, 2, \dots$ (k — номер итерации) вычисляют $(k+1)$ -е приближение к корню по формуле

$$x_{k+1} = x_k + \frac{|f(x_k)|}{M}. \quad (\text{V.47})$$

При этом проверяется условие $x_{k+1} < b$. Если оно не выполняется, то считают, что все корни найдены, в противном случае продолжают вычисления, проверяя неравенство

$$\frac{|x_{k+1} - x_k|}{|x_k|} \leq \frac{\varepsilon}{1 + \varepsilon} \frac{\inf_x |f'(x)|}{\sup_x |f'(x)|}.$$

При невыполнении неравенства повторяют вычисления по формуле (V.47), в противном случае полагают x_{k+1} одним из корней уравнения (V.21) на отрезке $[a, b]$.

Начальное приближение к следующему корню вычисляют по формуле

$$x_0 = (1 + \varepsilon) x_{k+1}.$$

Если $x_0 < b$, то вычисления повторяют, а если $x_0 \geq b$, то итерационный процесс считают законченным.

Суммарное число итераций для вычисления всех действительных корней на отрезке $[a, b]$ оценивается формулой

$$S < \frac{b-a}{\varepsilon} M.$$

На одной итерации в этом методе требуются одна операция умножения, одна операция сложения и ω_i операций для вычисления функции $f(x_k)$.

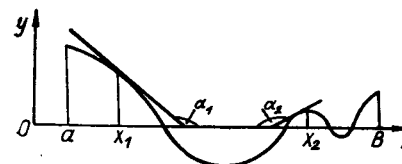


Рис. 15



Рис. 16

Пример 1. Найти действительные корни уравнения (V.33) на отрезке $[-4, 1]$.

Решение. Проверяем условия применимости метода. Функция в (V.33) непрерывна, а производная $f'(x) = 3x^2 + 6x$ ограничена. Поэтому для решения можно использовать метод Рыбакова, реализуемый по формуле (V.47), где $M = 24$. В результате решения на ЭВМ МИР-2 с шестью десятичными знаками получаем

$$x_1 = -2,879\,49, \quad x_2 = -0,653\,050, \quad x_3 = 0,531\,845.$$

Рассматриваемый метод допускает простую геометрическую интерпретацию (рис. 15).

В произвольной точке x величина $f'(x)$ есть тангенс угла наклона к оси x касательной к графику функции $y = f(x)$ в точке x , т. е. $f'(x) = \tan \alpha$. Так как угол α может быть как острым, так и тупым (см. рис. 15), значение $f'(x) = \tan \alpha$ может быть как положительным, так и отрицательным. Величина $|f'(x)| = |\tan \alpha|$ есть модуль тангенса угла наклона, а M в (V.46) — оценка сверху для наибольшего значения $|\tan \alpha| = |f'(x)|$ при всех $a \leq x \leq b$.

График функции $y = |f(x)|$ лежит выше оси x , так как та часть графика функции $y = f(x)$, которая лежит ниже оси x , зеркально отображается относительно ее (рис. 16).

Вначале задают начальное приближение к действительному корню $x_0 = a$. Через точку $(x_0, f(x_0))$ проводят прямую под углом к оси абсцисс, большим, чем угол любой из касательных к $f(x)$ на отрезке $[a, b]$. Пересечение полученной прямой с осью абсцисс дает первое приближение x_1 . Вычисление функции в геометрической интерпретации есть восстановление перпендикуляра к оси абсцисс в точке x_1 . Если решение не получено, то снова через точку $(x_1, f(x_1))$ проводят прямую, которая относительно оси абсцисс имеет тангенс угла наклона, равный M , и так далее, до тех пор пока не получим искомый корень x_{k+1} с заданной точностью. Затем, отступая от этого корня на εx_{k+1} , повторяем итерационную процедуру до получения следующего корня.

6. Метод секущих. При нахождении нулей функции f , для которой вычисление $f'(x)$ затруднено, метод секущих часто является лучшим, чем метод Ньютона. Этот алгоритм начинается с задания двух исходных чисел: x_1 и x_2 . На каждом шаге x_{k+1} получают из x_k и x_{k-1} как единственный нуль линейной функции, принимающей значения $f(x_k)$ в x_k и $f(x_{k-1})$ в x_{k-1} . Эта функция представляет собой секущую к кривой $y = f(x)$, проходящую через ее точки с абсциссами x_k и x_{k-1} . Отсюда и название — метод секущих.

Легко показать, что

$$x_{k+1} = x_k - \frac{f_k}{(f_{k-1} - f_k)/(x_{k-1} - x_k)}, \quad (V.48)$$

где $f_k = f(x_k)$.

Метод секущих относится к так называемым двухшаговым итерационным процессам (см. гл. II), и доказательство его сходимости не вписывается в схемы, рассмотренные в п. 4 настоящего параграфа [175].

Если для уравнения (V.21) $f(x^*) = 0$, однако $f'(x^*) \neq 0$, $f''(x^*) \neq 0$ и f'' непрерывна, то существует открытый интервал, содержащий корень x^* . Этот интервал такой, что если x_1 и x_2 — две различные точки из этого интервала, то последовательность, получаемая по формуле (V.48), сходится к x^* при $k \rightarrow \infty$, причем справедлива оценка [110]

$$|x_{k+1} - x^*| \leq c |x_k - x^*|^{1,618}, \quad (V.49)$$

где $c = \text{const}$. Таким образом, метод секущих имеет асимптотический показатель скорости сходимости $\alpha = 1,618$.

Сравнение метода секущих и простой итерации (см. п. 4 настоящего параграфа) по показателю сходимости приводит к заключению о большей эффективности по скорости сходимости метода секущих. Кроме того, эти методы сравнимы по числу арифметических операций ω_f , затрачиваемых на вычисление функций $f(x)$ на каждой итерации. Таким образом, метод секущих оказывается предпочтительней метода простой итерации.

Теперь сравним метод Ньютона (см. п. 4 настоящего параграфа) и метод секущих. Если допустим, что на вычисление $f(x)$ и $f'(x)$ затрачивается одинаковое число операций ω_f , то одна итерация по методу Ньютона $|x_{k+1} - x^*| \leq c |x_k - x^*|^2$ по числу операций эквивалентна двум итерациям по методу секущих

$$|x_{k+2} - x^*| \leq c_1 |x_{k+1} - x^*|^{1,618} \leq \tilde{c}_1 |x_k - x^*|^{2,618}, \quad (V.50)$$

$$\tilde{c}_1 = c_1^{2,618}.$$

При этих условиях метод секущих оказывается эффективнее ($2,618 > 2$) метода Ньютона. Отметим, что в отличие от метода Ньютона в методе секущих не нужно вычислять $f'(x)$.

Вычислительная схема этого метода может быть реализована следующим образом. До начала вычислений задают концы отрезка x_1, x_2 , на которых функция $f(x)$ имеет значения разных знаков, критерий окончания итерационного процесса

$$\frac{|x_{k+1} - x_k|}{|x_k|} \leq \frac{\varepsilon}{1 + \varepsilon} \frac{\inf_x |f'(x)|}{\sup_x |f'(x)|}, \quad (V.51)$$

где ε — относительная погрешность решения, полученного в ходе итерационного процесса.

Сначала полагают $x_{k-1} = x_1, x_k = x_2$, затем производят вычисление x_{k+1} по формуле (V.48). Проверяют условие окончания итераций (V.51).

Если оно не выполняется, то подсчитывают следующее приближение, а если выполняется, то считают

$$\frac{|x^* - x_{k+1}|}{|x^*|} \leq \varepsilon. \quad (V.52)$$

На одну итерацию в этом методе требуются три операции вычитания, одна операция умножения, одна операция деления и ω_f операций на вычисление функции $f(x)$.

Число итераций в этом методе приближенно определяется формулой

$$S \leq \frac{\ln \varepsilon - \ln(b-a)}{\ln q},$$

где

$$q = (b-a) \max_{a \leq x \leq b} \left| \frac{f''(x)}{f'(x)} \right|.$$

Пример 2. Методом секущих вычислить корень уравнения (V.33) на отрезке $[0,1; 1]$, применяя для окончания итерационного процесса условие (V.51), при $\varepsilon = 0,01$.

Решение. Проверяем условия применимости метода секущих. Функция в (V.33) непрерывна и на отрезке $[0,1; 1]$ имеет корень, так как $f(0, 1) f(1) < 0$. Найдем две первые производные $f(x)$:

$$f'(x) = 3x^2 + 6x,$$

$$f''(x) = 6x + 6.$$

Производные f' и f'' на отрезке $[0,1; 1]$ не меняют знака, следовательно, в нуль не обращаются, и функция $f''(x)$ непрерывна. Поэтому рассматриваемую задачу можно решать методом секущих.

Вычислим сначала правую часть условий окончания итерационного процесса (V.51):

$$\frac{\varepsilon}{1 + \varepsilon} \frac{\inf_x |f'(x)|}{\sup_x |f'(x)|} = \frac{0,01}{1,01} \frac{0,63}{9} \approx 6 \cdot 10^{-4}.$$

Положим $x_1 = 0,1, x_2 = 1$. Далее, используя формулу (V.48), вычисляем $x_3 = 0,319\,727\,9$.

Проверяем условие окончания итераций (V.51):

$$\frac{|x_3 - x_2|}{|x_2|} = 0,680\,272\,21 > 6 \cdot 10^{-4}.$$

Так как условие окончания итераций не выполнено, продолжаем итерационный процесс, вычисляя значения x_4, x_5 и другие и, наконец, для $x_8 = 0,532\,089\,2$ имеем

$$\frac{|x_8 - x_7|}{|x_7|} = 0,000\,202\,507 < 6 \cdot 10^{-4}.$$

Таким образом, приближенное решение $x = 0,532\,089\,2$.

7. Нахождение комплексных корней трансцендентных уравнений.

При решении прикладных задач могут возникать трансцендентные уравнения вида

$$f(z) = 0, \quad (V.53)$$

где $f(z)$ — непрерывная функция комплексной переменной. Теоретически функция $f(z)$ может быть сколь угодно точно приближена многочленами $P_n(z)$ достаточно большой степени (о задачах приближения функций см. гл. VI). Таким образом, иногда вместо решения уравнения (V.53) можно находить корни полиномиального уравнения

$$P_n(z) = 0. \quad (V.54)$$

Если функция $f(z)$ в (V.53) является аналитической функцией, имеющей не только действительные, но и комплексные корни, то можно использовать методы Ньютона и секущих, реализуемые в комплексных областях. Так как оба метода обладают локальной сходимостью, в случае получения расходящегося или заклинивающегося итерационного процесса целесообразно изменить, может быть, несколько раз начальное приближение.

Если найден один корень z_1 уравнения (V.53) и нужно найти другой, то целесообразно продолжить итерационный процесс с новой функцией

$$f_1(z) = \frac{f(z)}{z - z_1},$$

где деление выполняется лишь для числовых значений $f(z)$ и $z - z_1$. Основной член в методе Ньютона — отношение $\frac{f'(z)}{f(z)}$, обратная к нему величина $\frac{f'(z)}{f(z)} = \frac{d \ln f(z)}{dz}$. Поэтому

$$\frac{f_1'(z)}{f_1(z)} = \frac{d}{dz} \ln f(z) - \frac{d}{dz} \ln(z - z_1) = \frac{f'(z)}{f(z)} - \frac{1}{z - z_1}.$$

После того как найдено число корней m , метод Ньютона применяют к функции $\frac{f(z)}{\prod_{k=1}^m (z - z_k)}$ и в принципе логарифмическое дифференцирование производится легко.

Для получения хорошего начального приближения перед использованием метода Ньютона или секущих можно использовать алгоритм для минимизации действительной функции двух переменных

$$\varphi(x, y) = |f(x + iy)|^2, \quad i = \sqrt{-1}.$$

Отметим, что для аналитической функции f все локальные минимумы функции $\varphi(x, y)$ соответствуют случаю $\varphi = 0$.

Для решения уравнения (V.53) можно использовать метод Мюллера [56]. В отличие от метода секущих, где используется линейная интерполяция, в этом методе функцию $f(z)$ приближают по точкам z_j, z_{j-1}, z_{j-2} интерполяционным полиномом Ньютона второй степени (см. п. 5 параграфа VI.1). Существенным преимуществом метода Мюллера является то, что он сходится с любого начального приближения.

Предположим, что итерации начаты и найдены три точки:

$$(z_j f(z_j)), (z_{j-1}, f(z_{j-1})) \text{ и } (z_{j-2}, f(z_{j-2})).$$

Функцию $f(z)$ аппроксимируют следующим образом:

$$f(z) \approx P_2(z) = f(z_j) + (z - z_j)p + (z - z_j)(z - z_{j-1})q, \quad (V.55)$$

где

$$p = \frac{f(z_j) - f(z_{j-1})}{z_j - z_{j-1}}, \quad q = \frac{1}{z_j - z_{j-2}} \left[\frac{f(z_j) - f(z_{j-1})}{z_j - z_{j-1}} - \frac{f(z_{j-1}) - f(z_{j-2})}{z_{j-1} - z_{j-2}} \right]. \quad (V.56)$$

Корни полинома второй степени относительно z в (V.55) можно определить по формуле

$$z_{1,2} = z_j - \frac{p + (z_j - z_{j-1})q \pm \sqrt{(p + (z_j - z_{j-1})q)^2 - 4f(z_j)q}}{2q}.$$

Из корней z_1, z_2 выбирают тот, который ближе к z_j . Таким образом, получают приближение к корню. Итерационный процесс продолжают до тех пор, пока не будут выполнены условия окончания итераций.

Вычислительную схему метода Мюллера можно реализовать следующим образом. До начала вычислений задают начало и конец диаметра окружности, содержащей корни, $a + ib$ и $c + ie$ и величину ϵ_1 , по которой заканчивают итерации, а также некоторое малое положительное число ϵ_2 . Затем полагают

$$z_{j-2} = a + ib, \quad z_{j-1} = \frac{(a+c) + i(b+e)}{2}, \quad z_j = c + ie.$$

Вычисляют величины

$$v_j = \frac{f(z_j) - f(z_{j-1})}{z_j - z_{j-1}}, \quad q_j = \frac{1}{z_j - z_{j-2}}(v_j - v_{j-1}),$$

$$p = v_j + (z_j - z_{j-1})q, \quad d = \sqrt{p^2 - 4f(z_j)q},$$

$$z^1 = z_j - \frac{p+d}{2q}, \quad z^2 = z_j - \frac{p-d}{2q}.$$

Полагают $z_{j-2} = z_{j-1}, z_{j-1} = z_j$.

Если $|z^1 - z_j| < |z^2 - z_j|$, то $z_j = z^1$, иначе $z_j = z^2$. В случае если не выполняется условие окончания счета

$$|f(z_j)| < \epsilon_1,$$

то повторяют вычисления v, q, p, d, z^1, z^2 для следующей точки z_{j+1} , а если этот критерий выполняется, то осуществляется проверка полученного корня. Если $|\operatorname{Im}(z_j)| < \epsilon_2$, то мнимой частью корня можно пренебречь и считать его действительным.

Проверяют, все ли корни найдены. При необходимости нахождения следующих корней понижают предварительно степень уравнения, рассматривая

$$f_1(z) = \frac{f(z)}{z - z^*},$$

и повторяют вычисления в том же интервале.

Пример 3. Методом Мюллера решить уравнение

$$f(z) = z^3 - z^2 + z - 1 = 0. \quad (V.57)$$

Решение. Выбираем $\varepsilon_1 = 10^{-2}$ и начальное приближение $z_0 = 0,1$; $z_1 = 0,5$; $z_2 = 0,9$. Вычисляем $f(z_0) = -0,909$; $f(z_1) = -0,625$; $f(z_2) = -0,181$; $v_1 = 0,71$; $v_2 = 1,11$; $g_2 = 0,5$; $p_2 = 1,31$; $d_2 = 1,442$;

$$z^1 = 0,9 - \frac{1,31 + 1,442}{1} = -1,852,$$

$$z^2 = 0,9 - \frac{1,31 - 1,442}{1} = 1,032.$$

Таким образом, для следующего приближения имеем $z_1 = 0,5$; $z_2 = 0,9$; $z_3 = 1,032$. Далее имеем $f(z_1) = -0,625$; $f(z_2) = -0,181$; $f(z_3) = 0,066$; $v_2 = 1,11$; $v_3 = 1,871$; $q_3 = 1,430$; $p_3 = 2,06$; $d_3 = 1,966$; $z^1 = 1,032 - \frac{2,06 + 1,966}{2,86} = -0,3756923$; $z^2 = 1,032 - \frac{2,06 - 1,966}{2,86} = 0,9991329$. Так как $|z^1 - z_3| > |z^2 - z_3|$, то $z_4 = 0,9991329$. Учитывая, что $f(z_4) = -0,0017332$ и $|f(z_4)| < 10^{-2}$, корнем искомого уравнения является $z^* = 0,9991329$.

Разделив исходное уравнение на $z - z^*$, получим квадратное уравнение, из которого легко определить два оставшихся корня.

На одну итерацию в методе Мюллера требуются четыре операции умножения, шесть операций сложения, одна операция по вычислению квадратного корня и ω_i операций по вычислению $f(z)$.

8. Численное решение полиномиальных уравнений. Пусть требуется найти корни уравнения (V.54). Используя информацию о свойствах корней таких уравнений и их специфику, можно построить достаточно эффективные численные методы вычисления как действительных, так и комплексных корней.

Решение уравнений второй, третьей и четвертой степеней может быть получено в радикалах, которые можно найти в любом справочнике по математике. Несмотря на кажущуюся простоту этих формул машинная реализация их для всевозможных случаев задания коэффициентов далеко не тривиальная задача.

Одним из машинных способов решения квадратного уравнения

$$x^2 + px + q = 0 \quad (V.58)$$

может быть такой:

$$x_1 = \begin{cases} a + d, & a \geq 0, \\ a - d, & a < 0, \end{cases} \quad x_2 = \frac{q}{x_1}, \quad a = -\frac{p}{2}, \quad d = \sqrt{a^2 - q}.$$

В работе [110] отмечено, что в задаче нахождения корней уравнения (V.58) требуются различные преобразования основной формулы и предполагается наличие нескольких практических алгоритмов в зависимости от коэффициентов уравнения. В общем случае уравнения выше четвертой степени не допускают решения в радикалах, и поэтому их решают с помощью итерационных методов. Рассмотрим одну из возможных реализаций обобщенного метода Ньютона для решения уравнения (V.54).

До начала вычислений задают начальное приближение к корню

$$z_0 = (x_0, y_0) = x_0 + iy_0.$$

При использовании этого метода полагают, что функция $f(z) = P_n(z)$ имеет производные, по крайней мере, до l -го порядка включительно. Находят первую, отличную от нуля, производную функции $f(z)$ в точке z_0 , т. е.

$$f^{(l)}(z_0) \neq 0.$$

Итерационный процесс осуществляют по формуле

$$z_{k+1} = z_k + t \left(-\frac{f(z_k)}{f^{(l)}(z_k)} \right)^{\frac{1}{l}}. \quad (V.59)$$

Параметр t выбирают таким, чтобы выполнялось неравенство

$$|f(z_{k+1})| < |f(z_k)|. \quad (V.60)$$

Параметр t можно выбрать следующим образом. Полагают $t_0 = 1$, по z_k находят z_{k+1} . Затем проверяют условие (V.60). Если оно не выполняется, то полагают $t_{i+1} = \frac{t_i}{2}$, и снова по z_k находят z_{k+1} . Этот процесс уменьшения параметра t продолжают до тех пор, пока не выполнится неравенство (V.60). В этом случае полагают $z_k = z_{k+1}$ и продолжают итерационный процесс нахождения корня. Описанный метод сходится с любого начального приближения к одному из корней. Определив корень, степень многочлена можно понизить по схеме Горнера. Пусть z^* — корень уравнения

$$P_n(z) = a_0 z^n + a_1 z^{n-1} + \dots + a_n = 0,$$

тогда коэффициенты многочлена

$$P_{n-1}(z) = \frac{P_n(z)}{z - z^*} = b_0 z^{n-1} + b_1 z^{n-2} + \dots + b_{n-1}$$

определяются по рекуррентным формулам

$$b_0 = a_0, \quad b_j = a_j + b_{j-1} z^* \quad (j = 1, 2, \dots, n-1).$$

Таким образом определяются все корни многочлена.

Пример 4. Найти корни полиномиального уравнения (V.57) обобщенным методом Ньютона с точностью $\varepsilon = 0,01$.

Решение. В качестве начального приближения выбираем $z_1^0 = 0$. В этой точке определяем

$$P_3(z_1^0) = -1, \quad |P_3(z_1^0)| = 1.$$

Находим производную в точке z_1^0

$$P'_3(z_1^0) = (3z^2 - 2z + 1)|_{z=z_1^0} = 1.$$

Значит, $l = 1$. Далее получаем, что первое приближение к корню

$$z_1^1 = z_1^0 - \frac{P_3(z_1^0)}{P'_3(z_1^0)} = 0 - \frac{(-1)}{1} = 1.$$

Определяем

$$P_3(z_1^1) = 1^3 - 1^2 + 1 - 1 = 0.$$

Ясно, что $z_1^1 = 1$ является корнем уравнения. Уменьшаем степень исходного уравнения

$$P_2(z) = \frac{P_3(z)}{z-1} = \frac{z^3 - z^2 + z - 1}{z-1} = z^2 + 1.$$

Выбираем новое начальное приближение $z_2^0 = 0$ к следующему корню. Находим, что

$$P_2(z_2^0) = 1, |P_2(z_2^0)| = 1, P_2'(z) = 2z, |P_2'(z_2^0)| = 0.$$

Так как $P_2'(z_2^0) = 0$, находим следующую производную:

$$P_2''(z) = 2 \neq 0.$$

Значит, $l = 2, t = 1$. Подставляя эти значения в формулу (V.59), имеем

$$z_2^1 = z_2^0 + \left(-\frac{P_2(z_2^0)}{P_2''(z_2^0)} \right)^{1/2} = 0 + \left(-\frac{1}{2} \right)^{1/2} = +\frac{1}{\sqrt{2}}i.$$

(Нами был выбран положительный знак при извлечении квадратного корня.) Далее находим

$$P_2(z_2^1) = \left(\frac{i}{\sqrt{2}} \right)^2 + 1 = \frac{1}{2},$$

$$|P_2(z_2^1)| = \frac{1}{2} < 1 = |P_2(z_2^0)|.$$

Проводим следующую итерацию:

$$P_2'(z_2^1) = 2z_2^1 = \sqrt{2}i.$$

Ясно, что $l = 1, t = 1$. Находим следующее приближение:

$$z_2^2 = z_2^1 - \frac{P_2(z_2^1)}{P_2'(z_2^1)} = \frac{i}{\sqrt{2}} - \frac{1}{2\sqrt{2}i} = \frac{i}{\sqrt{2}} + \frac{i}{2\sqrt{2}} = \frac{3i}{2\sqrt{2}},$$

$$P_2(z_2^2) = \left(\frac{3i}{2\sqrt{2}} \right)^2 + 1 = -\frac{9}{8} + 1 = -\frac{1}{8},$$

$$|P_2(z_2^2)| = \frac{1}{8} < \frac{1}{2} = |P_2(z_2^1)|.$$

Переходим к нахождению следующего приближения:

$$P_2(z_2^2) = 2z_2^2 = \frac{3i}{\sqrt{2}},$$

значит

$$l = 1, t = 1 \text{ и}$$

$$z_2^3 = z_2^2 - \frac{P_2(z_2^2)}{P_2'(z_2^2)} = \frac{3i}{2\sqrt{2}} + \frac{1}{\frac{3i}{\sqrt{2}}} = \frac{3\sqrt{2}i}{4} - \frac{\sqrt{2}i}{24} = \frac{17\sqrt{2}}{24}i,$$

$$|P_2(z_2^3)| \approx 0,003 < \varepsilon.$$

В качестве приближенного значения корня можно принять

$$z_2 = \frac{17\sqrt{2}}{24}i \approx 1,0083i.$$

Так как алгебраическое уравнение четной степени с действительными коэффициентами имеет попарно сопряженные корни, то $z_3 = -1,0083i$. Значит, корнями уравнения будут: $z_1 = 1, z_2 = 1,0083i, z_3 = -1,0083i$.

9. Погрешность машинной реализации. В ходе реализации алгоритмов решения нелинейных уравнений на ЭВМ вычисления ведутся с ограниченным числом десятичных знаков. При этом в некоторых случаях даже малые погрешности, сравнимые с погрешностью округления, могут значительно исказить решение. В работе [97] рассматривается следующий пример. Уравнение

$$P_n(x) = \prod_{k=1}^{20} (x - k) = 0$$

имеет корни $x_k = k, k = 1, 2, \dots, 20$, а уравнение

$$\bar{P}_n(x) = \prod_{k=1}^{20} (x - k) - 2^{-23}x^{19},$$

в котором возмущение 2^{-23} в коэффициенте при x^{19} составляет приблизительно 2^{-30} его величины, имеет корни

1,000 000 000,	6,000 006 944,	10,095 266 145,	$\pm 0,643 500 904,$
2,000 000 000,	6,999 697 234,	11,793 633 881,	$\pm 1,652 329 728,$
3,000 000 000,	8,007 267 603,	13,992 858 137,	$\pm 2,518 830 070,$
4,000 000 000,	8,917 250 249,	16,730 737 466,	$\pm 2,812 624 894,$
4,999 999 928,	20,846 908 101,	19,502 439 400,	$\pm 1,940 330 347.$

Наблюдается высокая чувствительность наибольших корней к возмущению старших коэффициентов. Рассмотрим теоретически влияние ошибок машинной реализации на примере одношаговых итерационных процессов. Учет ошибок округлений чисел можно осуществить, вводя в итерационный процесс (V.35) множитель $1 + \delta_k$, т. е. рассматривая формулу

$$\bar{x}_{k+1} = \varphi(\bar{x}_k) = \varphi(\bar{x}_k)(1 + \delta_k),$$

где $\varphi(x)$ — сжимающая функция, x_k — фактически получаемая последовательность чисел с учетом ошибок округления, k — номер итерации.

Если x^* — точное решение математического уравнения, то

$$x^* = \varphi(x^*).$$

Вычитая это равенство из предыдущего, получаем

$$\bar{x}_{k+1} - x^* = \varphi(\bar{x}_k) - \varphi(x^*) + \delta_k \varphi(\bar{x}_k) = \varphi'(\theta)(\bar{x}_k - x^*) + \delta_k \varphi(\bar{x}_k),$$

где $\theta \in [\bar{x}_k, x^*]$. Так как $\varphi(x)$ — сжимающая функция, можно записать

$$|\bar{x}_{k+1} - x^*| \leq \alpha |\bar{x}_k - x^*| + |\delta_k| |\varphi(\bar{x}_k)|,$$

а следовательно,

$$|\bar{x}_{k+1} - x^*| \leq \alpha^{k+1} |\bar{x}_0 - x^*| + \sum_{i=1}^k \alpha^{k-i} |\delta_i \varphi(\bar{x}_i)|,$$

или

$$|\bar{x}_{k+1} - x^*| \leq \alpha^{k+1} |\bar{x}_0 - x^*| + \delta, \quad (V.61)$$

где

$$\delta = \sum_{i=1}^k \alpha^{k-i} |\delta_i \varphi(\bar{x}_i)|.$$

Из последнего неравенства видно, что при $\alpha < 1$ и $k \rightarrow \infty$ значение $\bar{x}_{k+1}$ может не стремиться к точному решению математической задачи.

Рассмотрим уравнение

$$f(x) = 0,001 - 100e^{-\left(10 + \frac{x}{5}\right)} = 0, \quad 0 \leq x \leq 10, \quad (V.62)$$

имеющее математическое решение

$$x = 5(5 \ln 10 - 10) = 7,565.$$

Для решения уравнения (V.62) можно использовать метод простой итерации, реализуемый по формуле

$$x_{k+1} = x_k - 0,001 + 100e^{-\left(10 + \frac{x_k}{5}\right)}, \quad (V.63)$$

так как выполнены достаточные условия сходимости этого метода:

$$|\varphi'(x)| = |1 - 20e^{-\left(10 + \frac{x}{5}\right)}| < 1, \quad 0 \leq x \leq 10.$$

Машинное решение уравнения (V.62) по формуле (V.63) на ЭВМ МИР-2 с длиной машинного слова четыре десятичных знака и условием окончания итераций по формуле (V.43) для $\varepsilon = 10^{-2}, 10^{-3}$ за 3145 итераций получено следующее: $x_k = 5,005$; относительная погрешность машинного решения составляет 33,8 %, в то время как математическое решение равно 7,565.

Увеличение длины машинного слова ЭВМ МИР-2 до восьми десятичных знаков и использование той же программы для $\varepsilon = 10^{-2}$ и 10^{-3} дают такие результаты: $x_k = 7,518\,856\,5$ (число итераций 21 954, относительная погрешность 4,6 %) и $x_k = 7,559\,634\,1$ (число итераций 33 237, относительная погрешность 0,53 %) соответственно. Учитывая изложенное выше и рассматриваемый пример, можно сделать вывод, что получаемое машинное решение x_k должно быть оценено с точки зрения близости его к математическому решению задачи. Теоретически для этой цели можно использовать формулу (V.61). Практически для оценки близости машинного и математического решений можно решить рассматриваемую задачу дважды с обычной и удвоенной длиной машинного слова или решить рассматриваемую задачу другим итерационным методом.

Так, для решения уравнения (V.62) на ЭВМ МИР-2 с длиной машинного слова четыре десятичных знака можно использовать метод

Ньютона, реализуемый по формуле

$$x_{k+1} = x_k - \frac{0,001 - 100e^{-\left(10 + \frac{x_k}{5}\right)}}{20e^{-\left(10 + \frac{x_k}{5}\right)}},$$

и условие окончания итераций (V.45) при $\varepsilon = 10^{-2}, 10^{-3}$. Результаты решения приведены в табл. 9. Начальным приближением

для всех итерационных процессов было значение $x_0 = 1$.

Накопление погрешности машинной реализации наблюдается не только при вычислении корней трансцендентных уравнений. Если для нелинейного алгебраического уравнения

$$P_n(x) = 0$$

на отрезке $[a, b]$ каким-либо численным методом найден один корень x_1 , то остальные корни обычно определяют из уравнения

$$P_{n-1}(x) = \frac{P_n(x)}{x - x_1} = 0.$$

Так как корень x_1 вычислен лишь приближенно, то при делении на $x - x_1$ получается некоторый остаток, т. е.

$$P_n(x) = P_{n-1}(x)(x - x_1) + r,$$

при этом корни $P_{n-1}(x)$ могут отличаться от корней $P_n(x)$. Пусть x_2 — корень $P_{n-1}(x)$. Подставляя $x = x_2$ в предыдущее равенство, получаем

$$P_n(x_2) = r.$$

Поэтому при уменьшении степени полинома необходимо проводить уточнение вычисленных корней хотя бы методом Липша — Берстоу.

В.3. РЕШЕНИЕ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

1. Предварительные замечания. Вычисление решения системы нелинейных алгебраических или трансцендентных уравнений — одна из сложных задач численного анализа, трудоемкость которой возрастает с ростом числа неизвестных. Пусть точное описание некоторого физического явления осуществляется с помощью системы нелинейных уравнений

$$\tilde{F}(x) = 0, \quad x \in D, \quad (V.64)$$

где $x = (x_1, x_2, \dots, x_n)$, $\tilde{F}(x) = (\tilde{f}_1(x), \tilde{f}_2(x), \dots, \tilde{f}_n(x))^T$.

Однако к сожалению есть не явное выражение $\tilde{F}$, а приближенное описание этого явления с помощью системы нелинейных уравнений:

$$F(x) = 0, \quad x \in D, \quad (V.65)$$

Таблица 9

Длина машинного слова в десятичных знаках	Относительная погрешность окончания итераций ε	Число итераций k	Машинное решение x_k	Относительная погрешность полученного решения, %
---	--	--------------------	------------------------	--

Четыре	10^{-2}	4	7,518	1,3
Четыре	10^{-3}	5	7,579	1,4
Восемь	10^{-2}	4	7,564 405 6	0,06
Восемь	10^{-3}	5	7,564 628 4	0,04

где

$$F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T.$$

Задана погрешность исходных данных

$$\|\tilde{F}(x) - F(x)\| = \|\delta F(x)\| \leq \varepsilon. \quad (V.66)$$

Будем предполагать, что, во-первых, решение математической задачи (V.65) существует, единственно, а, во-вторых, если для этой системы выполняется (V.66), то выполняется и неравенство

$$\|\tilde{x}^* - x^*\| \leq \delta, \quad (V.67)$$

где $\tilde{x}^*$ — вектор-решение системы (V.64), x^* — вектор-решение задачи (V.65), $\delta = \delta(\varepsilon) > 0$ — как угодно малое число.

В дальнейшем нам потребуется следующее утверждение.

Если $f: R^n \rightarrow R^n$ непрерывно дифференцируемо в R^n , причем $\|f'(x)^{-1}\| \leq \gamma$ для любого $x \in R^n$, то выполняется неравенство

$$\|x_1 - x_2\| \leq \gamma \|f(x_1) - f(x_2)\|, \quad (x_1, x_2 \in R^n). \quad (V.68)$$

Это утверждение можно получить как следствие из теоремы 5.3.11 работы [72].

Рассмотрим условия, при которых решение системы (V.65) можно считать решением системы (V.64). Для этого оценим норму разности решений систем (V.64) и (V.65) через погрешности в исходных данных и левую часть уравнения (V.65). Предположим, что функция F непрерывно дифференцируема и известна константа M :

$$\|F'(x)^{-1}\| \leq M, \quad x \in R^n. \quad (V.69)$$

Нетрудно убедиться, что справедливо соотношение

$$F(\tilde{x}^*) - F(x^*) = F(\tilde{x}^*) - \tilde{F}(x^*),$$

откуда

$$\|F(\tilde{x}^*) - F(x^*)\| = \|F(\tilde{x}^*) - \tilde{F}(x^*)\| = \|\delta F(\tilde{x}^*)\|. \quad (V.70)$$

Из (V.68) — (V.69) имеем

$$\|x^* - \tilde{x}^*\| \leq M \|\delta F(\tilde{x}^*)\|. \quad (V.71)$$

Назовем величину M в (V.69) локальным числом обусловленности функции $F(x)$, а величину

$$m = M \|\delta F(\tilde{x}^*)\| \quad (V.72)$$

числом обусловленности задачи. На практике будем исследовать такие задачи, для которых $m < 0,05$.

Следует отметить, что если имеется сходящийся итерационный процесс для решения (V.65), известно M в (V.69), x_k — приближение к решению, то

$$\frac{\|x_k - x^*\|}{\|x^*\|} \leq \varepsilon,$$

если $\frac{M \|F(x_k)\|}{\|x_k\| - M \|F(x_k)\|} \leq \varepsilon$ при $\|x_k\| > M \|F(x_k)\|$.

Плохая обусловленность систем нелинейных уравнений требует при постановке задачи, описываемой этой системой, дополнительного анализа с точки зрения физики исследуемых процессов, является ли полученное математическое решение решением той физической модели, которая исследуется.

Из оценки (V.72) следует, что в зависимости от поведения функции $F(x)$ в районе корня различные решения системы требуют различной точности исходных данных. Задавая одну и ту же точность исходных данных для всех решений системы, можно получить некоторые решения задачи (V.65), (V.66), отличающиеся от решения системы (V.64).

При анализе систем нелинейных уравнений не существует достаточно простых способов определения числа корней и их локализации, поэтому в прикладных задачах, описываемых системами нелинейных уравнений, отделение корней и выбор начального приближения иногда можно осуществить с использованием специфических особенностей поставленной задачи в соответствии с физикой процесса.

Задачу нахождения решения системы нелинейных уравнений можно свести к задаче минимизации некоторой функции. В последние годы методы минимизации функций как без ограничений, так и с ограничениями нашли широкое развитие.

В дальнейшем будем предполагать, что решение системы нелинейных уравнений (V.65) существует и единственно, а $F(x)$ обладает всеми необходимыми свойствами.

2. Метод простой итерации. Одним из наиболее простых методов решения систем нелинейных уравнений является метод простой итерации (см. п. 3 параграфа V.2), который для системы (V.65) может быть реализован в виде

$$x_{k+1} = \Phi(x_k), \quad (V.73)$$

где

$$\Phi(x_k) = x_k - F(x_k). \quad (V.74)$$

Как и прежде, формула (V.73) порождает итерационный процесс, условием сходимости которого является сжатость отображения (V.73), т. е. выполнение неравенства

$$\|\Phi(x) - \Phi(y)\| \leq \alpha \|x - y\|, \quad 0 < \alpha < 1, \quad (V.75)$$

для всех $x, y \in D_0$. В (V.75) отображение $\Phi(x)$ определено на множестве $D \subset R^n \rightarrow R^n$. Если $\Phi(x): D \subset R^n \rightarrow R^n$ — сжимающее на компактном множестве $D_0 \subset D$ отображение и $\Phi D_0 \subset D_0$, то $\Phi(x)$ имеет единственную неподвижную точку в D_0 , т. е. существует такая точка x^* , что

$$x^* = \Phi(x^*). \quad (V.76)$$

Для доказательства выберем произвольную точку $x_0 \in D_0$ и образуем последовательность

$$x_{k+1} = \Phi(x_k), \quad k = 0, 1, 2, \dots \quad (V.77)$$

Так как $\Phi D_0 \subset D_0$, последовательность $\{x_k\}$ лежит в множестве D_0 .

По условию $\Phi(x_k)$ — сжимающее отображение, поэтому

$$\|x_{k+1} - x_k\| = \|\Phi(x_k) - \Phi(x_{k-1})\| \leq \alpha \|x_k - x_{k-1}\|. \quad (V.78)$$

Далее имеем

$$\|x_{k+p} - x_k\| \leq \sum_{i=1}^p \|x_{k+i} - x_{k+i-1}\| \leq (\alpha^{p-1} + \dots + 1) \|x_{k+1} - x_k\| \leq \frac{\alpha^k}{1-\alpha} \|x_1 - x_0\|. \quad (V.79)$$

Оценка (V.79) показывает, что последовательность $\{x_k\}$ образует последовательность Коши. И так как D_0 — компактное множество, то последовательность Коши имеет предельную точку $x^* \in D_0$. Из непрерывности $\Phi(x)$ следует, что

$$\lim_{k \rightarrow \infty} \Phi(x_k) = \Phi(x^*)$$

и, значит, x^* — неподвижная точка.

Для доказательства единственности этой точки предположим противное, т. е., что существуют, по крайней мере, две неподвижные точки $x^* \neq x^{**}$, для которых выполняются равенства

$$x^* = \Phi(x^*), \quad x^{**} = \Phi(x^{**}).$$

Так как функция $\Phi(x)$ — сжимающая и $\alpha < 1$, то

$$\|x^* - x^{**}\| = \|\Phi(x^*) - \Phi(x^{**})\| \leq \alpha \|x^* - x^{**}\| < \|x^* - x^{**}\|.$$

Полученное противоречие доказывает единственность неподвижной точки.

Из приведенных рассуждений ясно, что метод простой итерации (V.73), (V.74) будет сходящимся, если функция

$$\Phi(x) = x - F(x) \quad (V.80)$$

будет сжимающей. Предположив, что функция $F(x)$ непрерывно дифференцируема, находят

$$\begin{aligned} \|\Phi(x) - \Phi(y)\| &= \|x - F(x) - y + F(y)\| \leq \\ &\leq \|x - y - B(x, y)(x - y)\| \leq \|E - B(x, y)\| \|x - y\|, \end{aligned} \quad (V.81)$$

где

$$B(x, y) = \begin{pmatrix} f'_1(x + t_1(x - y)) \\ \vdots \\ f'_n(x + t_n(x - y)) \end{pmatrix}, \quad t_1, t_2, \dots, t_n \in (0, 1).$$

Из оценки (V.81) следует, что функция $\Phi(x)$ будет сжимающей, если

$$\alpha = \max_{x, y \in D} \|E - B(x, y)\| < 1. \quad (V.82)$$

Таким образом, при выполнении условия (V.82) метод простой итерации будет сходящимся и обладать линейной скоростью сходимости. Однако это условие трудно проверить. Более простым для проверки условий сходимости метода простой итерации является условие

$$\left| \frac{\partial f_j}{\partial x_1} \right| + \left| \frac{\partial f_j}{\partial x_2} \right| + \dots + \left| \frac{\partial f_j}{\partial x_n} \right| < 1, \quad j = 1, 2, \dots, n, \quad x \in D,$$

где

$$\varphi_j(x_1, x_2, \dots, x_n) = x_j - f_j(x_1, x_2, \dots, x_n).$$

Если выполнены условия сходимости метода простой итерации, то оканчивать итерационный процесс можно при выполнении неравенства

$$\frac{\|x_{k+1} - x_k\|}{\|x_k\|} \leq (1 - q) \frac{\varepsilon}{1 + \varepsilon}, \quad (V.83)$$

где ε — требуемая «относительная» погрешность решения, q — некоторая константа, для которой $\|\Phi'(x)\| \leq q < 1, \forall x \in D$. Выполнение неравенства (V.83) гарантирует неравенство

$$\frac{\|x^* - x_k\|}{\|x^*\|} \leq \varepsilon. \quad (V.84)$$

Доказательство этого неравенства приводится в работе [68].

Для расширения области сходимости метода простой итерации в формулу (V.73), (V.74) вводят параметр $t \leq 0$ и реализуют итерации по формуле

$$x_{k+1} = x_k - tF(x_k). \quad (V.85)$$

Для сходимости метода (V.85) достаточно, чтобы матрица Якоби была знакоопределенной, а параметр t удовлетворял неравенству

$$|t| \leq \frac{1}{\max_{x \in D} \|F'(x)\|}.$$

Вычислительная схема метода простой итерации с параметром может быть реализована следующим образом. До начала вычислений задают начальное приближение x_0 , относительную погрешность решения ε и малое число $\delta > 0$ для окончания итерационного процесса, если параметр t очень мал.

Полагают $x_k = x_0$, а на последующих итерациях

$$x_k = x_{k+1}, \quad k = 0, 1, \dots, \quad (V.86)$$

и вычисляют

$$\psi(x_k) = (F(x_k), F(x_k)).$$

Положив $t = 1$, вычисляют

$$x = x_k - tF(x_k) \quad (V.87)$$

и проверяют выполнение условия

$$\psi(x) < \psi(x_k). \quad (V.88)$$

При выполнении условия (V.88) считают $x_{k+1} = x$. И если $\psi(x_{k+1}) > \varepsilon$, то итерационный процесс повторяют с (V.86). Если $\psi(x_{k+1}) < \varepsilon$, то проверяют условие (V.83), а если оно не выполнено, продолжают итерации. Если условие (V.83) выполнено, то считают, что получено решение с точностью ε (выполнено условие (V.84)). При невыполнении условия (V.88) проверяют условие

$$|t| < \delta. \quad (V.89)$$

Если оно не выполняется и $t > 0$, то $t = -t$, если же $t < 0$, то $t = -t/2$. Затем счет повторяют, начиная с формулы (V.87), при новом значении параметра t . В случае выполнения условия (V.89) считают, что процесс не сходится. На одну итерацию в этом методе (без параметра) требуются $2n$ операций сложения, ω операций для вычисления левых частей системы уравнений. Для каждой внутренней итерации с параметром t требуются n операций сложения, n операций умножения и ω операций по вычислению левых частей системы уравнений.

3. Метод Ньютона. По аналогии с функциями одной переменной (см. п. 3 параграфа V.2 (V.41)) для решения системы нелинейных уравнений (V.65) можно записать итерационную формулу метода Ньютона

$$x_{k+1} = x_k - [F'(x_k)]^{-1} F(x_k). \quad (V.90)$$

При использовании метода Ньютона вместо деления на производную $F'(x_k)$ в (V.41) на каждой итерации требуется обращение матрицы $F'(x_k)$, которое обычно заменяют решением системы линейных алгебраических уравнений

$$F(x_k) + F'(x_k)(x_{k+1} - x_k) = 0.$$

Поскольку часто по ряду причин трудно получить в явном виде матрицу Якоби $F'(x_k)$, при численной реализации метода используется ее конечно-разностная аппроксимация, например

$$(F'(x_k))_{ij} \cong \frac{f_i(x_k + h e_j) - f_i(x_k)}{h},$$

где h — шаг дифференцирования, e_j — j -й единичный орт.

При использовании этого метода предполагают, что в некоторой области D , содержащей решение x^* системы (V.65), функции $f_i(x)$ имеют непрерывные производные первого порядка и в некоторой окрестности точки x^* матрица Якоби $F'(x)$ (V.17) невырождена. В этом случае метод Ньютона обладает сверхлинейной скоростью сходимости. Если $F'(x)$ в некоторой окрестности корня x^* удовлетворяет условию

$$\|F'(x) - F'(x^*)\| \leq L \|x - x^*\|,$$

то наблюдается квадратичная сходимость.

Недостатком метода Ньютона является то, что если начальное приближение x_0 находится далеко от решения x^* , то метод чаще всего расходится. Существуют модификации метода Ньютона вида

$$x_{k+1} = x_k + t_k p_k, \quad (V.91)$$

где так называемое ньютоновское направление

$$p_k = -[F'(x_k)]^{-1} F(x_k), \quad (V.92)$$

значение $t_k > 0$ — некоторая длина шага вдоль p_k . Если функции $f_i(x)$ непрерывны и выполняется неравенство $\|F'(x)^{-1}\| \leq C$ для любых $x \in R^n$, то гарантируется сходимость к решению x^* из любой начальной точки $x_0 \in R^n$. При этом вблизи от x^* модификации полностью совпадают с традиционным методом Ньютона, что обеспечивает высокую скорость сходимости. Одна из первых таких модификаций

предложена в работе [79], а их наиболее общая схема дана в работе [9]. В них используется известное (см., например, [72]) свойство ньютоновского направления p_k , заключающееся в том, что при всех достаточно малых $t > 0$ (независимо от вида нормы $\|\cdot\|$) выполняется неравенство

$$\|F(x_k + t p_k)\| \leq \|F(x_k)\|, \quad x_k \neq x^*,$$

т. е. p_k является направлением убывания нормы невязки $\|F(x)\|$.

Опишем одну из модификаций. Она имеет некоторый параметр σ . Сходимость из любой начальной точки x_0 сохраняется при любых значениях $\sigma \in (0, 1)$, причем этот параметр слабо влияет на характер сходимости. Чаще всего используется $\sigma = 0,1$. Пусть имеется ньютоновское направление p_k . Положим $t_k = 1$ и проверим неравенство

$$\|F(x_k + t_k p_k)\| \leq (1 - \sigma t_k) \|F(x_k)\|, \quad (V.93)$$

где $\|\cdot\|$ — евклидова норма. Если оно выполняется, то переходим к (V.91), иначе дробим длину шага (полагая, например, $t_k = \frac{t_k}{2}$, до тех пор пока не выполнится (V.93)), и затем переходим к (V.91). Согласно (V.93) норма невязки $\|F(x_k)\|$ монотонно уменьшается с ростом k .

Отметим, что приведенная модификация незначительно отличается по трудоемкости от метода Ньютона, требуя на каждой итерации лишь вычисления значения F в нескольких дополнительных точках. При этом вблизи от решения неравенство (V.93) выполняется при $t_k = 1$ и дополнительные вычисления не требуются.

Критерием окончания итераций для метода Ньютона и его модификаций может служить неравенство

$$\|F(x_k)\| \leq \varepsilon, \quad (V.94)$$

где ε — заранее заданное положительное число.

Если получение матрицы Якоби требует сравнительно небольших вычислительных затрат, то метод Ньютона очень эффективен. В случае, когда вычислительные затраты, необходимые для нахождения матрицы Якоби, являются значительными, то вместо метода Ньютона используются его различные модификации, отличные от приведенной выше.

4. Методы квазиньютоновского типа. Одним из недостатков метода Ньютона является необходимость вычислять матрицу Якоби и решать систему линейных алгебраических уравнений. Это требует значительных затрат машинных операций, объем которых резко возрастает с ростом размерности системы (V.65). Поэтому были разработаны модификации метода Ньютона, в которых в ходе итерационного процесса вместо построения самой матрицы Якоби или ее обратной строится их аппроксимация. Это позволяет существенно сократить число арифметических операций на итерации. Такие методы решения систем нелинейных уравнений получили название квазиньютоновских. Показано [134, 142, 143], что большинство известных квазиньютоновских методов сходится локально со сверхлинейной скоростью сходимости при тех же предположениях о свойствах функций $f_i(x)$, которые были сделаны при

использовании метода Ньютона, имеющего квадратичную скорость сходимости (см. п. 3 настоящего параграфа). Достаточно полный обзор квазиньютоновских методов имеется в работах [143, 144]. Квазиньютоновские методы можно разбить на два тесно связанных между собой класса методов в зависимости от того, что аппроксимируется — матрица Якоби или обратная ей.

Рассмотрим сначала первый из классов, где матрица B_k с размерами $n \times n$ аппроксимирует матрицу $F'(\mathbf{x}_k)$. Перед началом итераций задают начальную точку $\mathbf{x}_0$, а матрицу B_0 обычно получают, либо полагая ее единичной, либо аппроксимируя $F'(\mathbf{x}_0)$ по конечно-разностным формулам. Затем для $k = 0, 1, \dots$ вычисляют

$$\mathbf{x}_{k+1} = \mathbf{x}_k - B_k^{-1} F(\mathbf{x}_k), \quad (\text{V.95})$$

$$\Delta \mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k,$$

$$\Delta F_k = F(\mathbf{x}_{k+1}) - F(\mathbf{x}_k),$$

$$B_{k+1} = B_k + \frac{(\Delta F_k - B_k \Delta \mathbf{x}_k) \mathbf{c}_k^T}{(\Delta \mathbf{x}_k^T \mathbf{c}_k)}, \quad (\text{V.96})$$

где $\mathbf{c}_k$ — n -мерный вектор, являющийся параметром рассматриваемого класса методов. Если $\mathbf{c}_k$ взять равным $\Delta \mathbf{x}_k$, то получится первый метод Бройдена [131]. Выбор $\mathbf{c}_k = \Delta F_k$ соответствует методу Пирсона [177], а $\mathbf{c}_k = \Delta F_k - B_k \Delta \mathbf{x}_k$ — симметричному методу первого ранга [132]. Возможны и другие способы выбора $\mathbf{c}_k$, которые приводят, например, к методу секущих [10, 132], являющемуся обобщением на случай функций многих переменных одномерного метода секущих (см. п. 6 параграфа V.2).

Во втором из рассматриваемых здесь классов квазиньютоновских методов матрица H_k с размерами $n \times n$ аппроксимирует матрицу $F'(\mathbf{x}_k)^{-1}$. Перед началом итераций задают начальную точку $\mathbf{x}_0$ и матрицу H_0 , которая обычно либо равна единичной, либо является обратной к конечно-разностной аппроксимации матрицы $F'(\mathbf{x}_0)$. Затем для $k = 0, 1, \dots$ вычисляют

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H_k F(\mathbf{x}_k), \quad (\text{V.97})$$

$$H_{k+1} = H_k + \frac{(\Delta \mathbf{x}_k - H_k \Delta F_k) \mathbf{d}_k^T}{(\Delta F_k^T \mathbf{d}_k)}, \quad (\text{V.98})$$

где $\mathbf{d}_k$ — n -мерный вектор, являющийся параметром рассматриваемого класса методов. Конкретный вид вектора $\mathbf{d}_k$ отвечает определенному методу. Например, $\mathbf{d}_k = \Delta F_k$ соответствует второму методу Бройдена [131], $\mathbf{d}_k = \Delta \mathbf{x}_k$ — методу Мак — Кормика (см. [177]).

Заметим, что, задав B_0^{-1} , можно вести пересчет не B_k , а матриц B_k^{-1} по формуле

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(\Delta \mathbf{x}_k - B_k^{-1} \Delta F_k) \mathbf{c}_k^T B_k^{-1}}{(\mathbf{c}_k^T B_k^{-1} \Delta F_k)}, \quad (\text{V.99})$$

эквивалентной (V.96). Это требует порядка $O(n^2)$ арифметических операций вместо $O(n^3)$ операций, необходимых для решения системы ли-

нейных уравнений

$$B_k \Delta \mathbf{x}_k = -F(\mathbf{x}_k).$$

Как видно из (V.99), между формулами (V.96) и (V.98) имеется определенная связь. Так, если $H_k = B_k^{-1}$, то $H_{k+1} = B_{k+1}^{-1}$ при $\mathbf{c}_k = B_k^{-1} \mathbf{d}_k$. Таким образом, один и тот же метод может быть реализован двумя различными формулами: (V.96) и (V.98), которые эквивалентны теоретически, но их численная реализация может отличаться по эффективности.

Рассмотрим, например, первый метод Бройдена. Его можно реализовать (см. [151]) по формуле (V.96) так, что это потребует всего $O(n^2)$ арифметических операций. Это оказывается возможным благодаря представлению матрицы B_k в виде произведения $Q_k R_k$, где Q_k — ортогональная, а R_k — верхняя треугольная матрица. Действительно, в этом случае решение системы требует только $O(n^2)$ арифметических операций, а, имея $B_k = Q_k R_k$, на представление матрицы B_{k+1} , удовлетворяющей (V.96), в виде $Q_{k+1} R_{k+1}$, необходимо затратить $O(n^2)$ арифметических операций. Здесь важное преимущество формулы (V.96) перед (V.98) заключается в том, что в (V.96) нет необходимости в умножении матрицы на вектор, поскольку

$$\Delta F_k - B_k \Delta \mathbf{x}_k = F(\mathbf{x}_{k+1}).$$

Довольно часто возникает необходимость в решении систем уравнений, у которых матрица Якоби $F'(\mathbf{x})$ — симметричная. Такие системы появляются при поиске стационарной точки модифицированных функций Лагранжа [181] для минимизации функций при наличии ограничений, а также в других важных задачах. Существуют квазиньютоновские методы, которые учитывают симметричность матрицы Якоби и вырабатывают последовательность симметричных матриц B_k (или H_k). Эти методы тоже можно разбить на два класса. В первом из них матрица B_k аппроксимирует $F'(\mathbf{x})$. В отличие от описанного выше класса, задаваемого формулами (V.97) и (V.96), здесь требуется симметричность матрицы B_0 , а вместо (V.96) используется формула

$$J_{k+1} = J_k + \frac{(\Delta F_k - B_k \Delta \mathbf{x}_k) \mathbf{c}_k^T - \mathbf{c}_k (\Delta F_k - B_k \Delta \mathbf{x}_k)^T}{(\Delta \mathbf{x}_k^T \mathbf{c}_k)} - \frac{((\Delta F_k - B_k \Delta \mathbf{x}_k)^T \Delta \mathbf{x}_k) \mathbf{c}_k \mathbf{c}_k^T}{(\Delta \mathbf{x}_k^T \mathbf{c}_k)^2}, \quad (\text{V.100})$$

где значение параметра $\mathbf{c}_k = \Delta \mathbf{x}_k$ соответствует симметричному варианту Пауэлла [181] метода Бройдена, а $\mathbf{c}_k = \Delta F_k$ — методу Давидона — Флетчера — Пауэлла [141, 149].

Во втором из рассматриваемых классов квазиньютоновских методов матрица H_k аппроксимирует матрицу $F'(\mathbf{x}_k)^{-1}$. Здесь матрица H_0 должна быть симметричной, а вместо (V.98) используется формула

$$H_{k+1} = H_k + \frac{(\Delta \mathbf{x}_k - H_k \Delta F_k) \mathbf{d}_k^T + \mathbf{d}_k (\Delta \mathbf{x}_k - H_k \Delta F_k)^T}{(\Delta F_k^T \mathbf{d}_k)} - \frac{((\Delta \mathbf{x}_k - H_k \Delta F_k)^T \Delta F_k) \mathbf{d}_k \mathbf{d}_k^T}{(\Delta F_k^T \mathbf{d}_k)^2}, \quad (\text{V.101})$$

где $d_k = \Delta x_k$ соответствует методу Бroyдена — Флетчера — Гольдфарба — Шенно [133, 148, 153, 188]. Этот метод является одним из лучших с вычислительной точки зрения, учитывающий симметричность матрицы Якоби.

Отметим, что формула (V.100) эквивалентна формуле

$$\begin{aligned} B_{k+1}^{-1} = & B_k^{-1} + [(c_k^T B_k^{-1} \Delta F_k) (B_k^{-1} c_k (\Delta x_k - B_k^{-1} \Delta F_k)^T + \\ & + (\Delta x_k - B_k^{-1} \Delta F_k) c_k^T B_k^{-1}) - (\Delta F_k^T (\Delta x_k - B_k^{-1} \Delta F_k)) B_k^{-1} c_k c_k^T B_k^{-1} + \\ & + (c_k^T B_k^{-1} c_k) (\Delta x_k - B_k^{-1} \Delta F_k) (\Delta x_k - B_k^{-1} \Delta F_k)^T] / [(c_k^T B_k^{-1} \Delta F_k)^2 + \\ & + (c_k^T B_k^{-1} \Delta F_k) (\Delta F_k^T (\Delta x_k - B_k^{-1} \Delta F_k))], \end{aligned}$$

которая исключает необходимость решения системы линейных уравнений. Что касается связи между формулами (V.100) и (V.101), то они (в отличие от формул (V.96) и (V.98)) не эквивалентны.

Описанные выше квазиньютоновские методы сходятся лишь при достаточно хорошем начальном приближении x_0 . Для расширения области их сходимости можно использовать прием, который называется одномерным поиском.

Пусть имеется квазиньютоновское направление $p_k = -B_k^{-1} F(x_k)$ (или $p_k = -H_k F(x_k)$). Примем длину шага $t_k = 1$ и проверим неравенство

$$\|F(x_k + t_k p_k)\| \leq \|F(x_k)\|, \quad (V.102)$$

где $\|\cdot\|$ евклидова норма. Если оно выполняется, то заканчиваем одномерный поиск и полагаем

$$x_{k+1} = x_k + t_k p_k, \quad (V.103)$$

т. е. дробим длину шага (полагая, например, $t_k = \frac{t_k}{2}$), до тех пор пока не выполнится (V.102). На этом заканчиваем одномерный поиск и переходим к формуле (V.103).

Видно, что одномерный поиск (если он удачен) обеспечивает монотонное уменьшение нормы невязки $\|F(x_k)\|$ с ростом k . Если квазиньютоновское направление p_k сильно отличается от ньютоновского (V.92), то одномерный поиск может оказаться неудачным, и тогда необходимо обновить матрицу B_k (или H_k), положив ее равной, например, конечно-разностной аппроксимации матрицы Якоби $F'(x_k)$ (или $F'(x_k)^{-1}$). Критерием окончания итераций для квазиньютоновских методов может служить неравенство (V.94).

Решение нелинейных алгебраических и трансцендентных уравнений является одной из классических задач численного анализа. Использование того или иного метода для решения этой задачи зависит от ее постановки, свойств входящих в нелинейное уравнение функций и имеющихся вычислительных средств. Задача нахождения корней нелинейного уравнения состоит из локализации корней, вычисления одного корня с заданной степенью точности.

В зависимости от свойств рассматриваемой нелинейной функции (трансцендентная или нелинейная алгебраическая, с действительными или комплексными коэффициентами и т. д.), постановки задач (определения действительных и комплексных корней, всех или заданных на определенном интервале и т. д.) и имеющейся электронной вычислительной машины (наличие дисплея, графопостроителей и других средств общения человека с машиной) применяют различные способы локализации корней (графические, табличные и др.).

В локализации корней широко используют результаты, полученные в высшей алгебре и математическом анализе (см., например, [116]).

Для вычисления действительных корней трансцендентных или нелинейных алгебраических уравнений можно использовать метод половинного деления, метод секущих, метод Ньютона и метод нахождения всех действительных корней на отрезке.

Если левая часть нелинейного алгебраического или трансцендентного уравнения непрерывна на рассматриваемом отрезке и на концах его имеет разные знаки, то для вычисления решения можно использовать метод половинного деления, оценив предварительно объем вычислительной работы. Если этот объем окажется слишком велик для имеющейся ЭВМ, то следует уменьшить длину отрезка.

Если функция, входящая в левую часть нелинейного уравнения, обладает некоторой гладкостью и первые, а также вторые производные от функции не меняют знака на рассматриваемом отрезке, то для решения такой задачи целесообразно использовать метод секущих.

В случае, когда первая производная от нелинейной функции существует на всем рассматриваемом отрезке и легко вычисляется, то для решения такой задачи можно использовать метод Ньютона, обладающий для некоторых корней квадратичной сходимостью.

Если первая производная от нелинейной функции может претерпевать разрывы первого рода, то для решения нелинейного уравнения с такой функцией можно использовать метод, позволяющий определять все действительные корни на отрезке.

При решении трансцендентных уравнений можно поступать двояким образом. Непрерывную функцию комплексной переменной можно сколь угодно близко аппроксимировать многочленом, а затем искать решение полученного нелинейного алгебраического уравнения. Если же в трансцендентном уравнении левая часть представляет собой аналитическую функцию, то для нахождения решения можно использовать методы секущих или Ньютона, реализуемые в комплексных областях, а также метод Мюллера.

Для нахождения корней полиномиальных уравнений полезным оказывается один из вариантов метода Ньютона. Для уточнения корней нелинейных алгебраических уравнений с действительными коэффициентами можно использовать метод Лина — Берстоу.

Численное решение систем нелинейных алгебраических или трансцендентных уравнений представляет собой сложную и до конца не решенную задачу вычислительной математики. Для решения систем нелинейных уравнений можно использовать метод простой итерации, метод секущих, метод Ньютона, квазиньютоновские методы и др. Простая

итерация обладает линейной сходимостью, метод Ньютона — квадратичной (при выполнении соответствующих условий), а квазиньютоновские методы — сверхлинейной скоростью сходимости. Несмотря на то что квазиньютоновские методы обладают худшей по сравнению с методом Ньютона теоретической сходимостью, они требуют при своей реализации меньшего числа машинных операций и во многих случаях предпочтительней классического метода Ньютона. Однако все эти методы обладают локальной сходимостью, т. е. сходимостью при хорошем начальном приближении. Для получения этого хорошего начального приближения при решении систем нелинейных уравнений используют те или иные методы спуска. И комбинируют методы спуска с методами, обладающими более высокой скоростью сходимости.

Отметим, что решение нелинейного уравнения или системы нелинейных уравнений может быть сведено к задаче минимизации функции. Задача поиска минимума функции n переменных является частным случаем решения экстремальных задач, т. е. задач на поиск минимумов, максимумов и так называемых седловых точек в минимаксных задачах.

Задачи нахождения экстремумов функции многих переменных естественным образом делятся на задачи нахождения безусловного экстремума, т. е. поиска экстремума функций на открытом множестве $D \subset R^n$, задачи нахождения экстремумов функций в областях D , определяемых какими-либо ограничениями.

Большинство методов безусловной минимизации функции многих переменных дают возможность находить локальный минимум, и лишь априорная информация о свойствах этой функции позволяет считать этот минимум глобальным. Методы, гарантирующие поиск глобального минимума с заданной точностью для достаточно общих классов функций, являются весьма трудоемкими. К ним относятся методы Монте-Карло, комбинация методов Монте-Карло определения начальной точки с одним из алгоритмов локального поиска, методы, основанные на построении нижней огибающей данной функции, методы последовательного отсечения подмножества, методы перебора на неравномерной сетке и т. д.

Для решения задач минимизации функции $F(x)$, обладающей непрерывными частными производными до r -го порядка включительно, широко используются методы спуска, реализуемые по формулам

$$x_{k+1} = x_k + \alpha_k p_k,$$

где α_k — действительное число, которое называют шагом множителем, вектор p_k определяет направление спуска. Среди методов спуска выделяют релаксационные, в которых

$$F(x_k) > F(x_{k+1})$$

при $k = 0, 1, 2, \dots$. Если $F(x)$ — непрерывно дифференцируемая функция, то релаксационность метода обеспечивается, когда направление спуска p_k образует острый угол с направлением антиградиента и α_k достаточно мал. В качестве основных параметров, характеризующих процесс, рассматриваются углы релаксации θ_k (углы между p_k и направлением градиента), а также множители релаксации q_k , определяемые

равенством

$$1 - \frac{q_k}{2} = \frac{F(x_k) - F(x_{k+1})}{(F'(x_k), x_k - x_{k-1})},$$

где F' — градиент функции F . Обозначим через

$$x_k = q_k(2 - q_k) \cos^2 \theta_k$$

приведенный коэффициент релаксации. Необходимое и достаточное условие сходимости релаксационного процесса для сильно выпуклой функции $F(x)$ ¹⁸:

$$\sum_{k=0}^{\infty} x_k = \infty.$$

Среди релаксационных методов наиболее известны градиентные. Их обычно делят на одно- и двухшаговые методы. Общая схема одношаговых градиентных методов реализуется следующим образом:

$$x_{k+1} = x_k - \alpha_k A(x_k) F'(x_k),$$

где α_k — шаговый множитель. Если в этой схеме $A(x_k) = E$, $\alpha_k = \text{const}$, E — единичная матрица, то получают градиентный спуск с постоянным шагом. Если в рассматриваемой схеме $A(x_k) = E$ и α_k определяется из условия минимума $F(x_k - \alpha_k F'(x_k))$, то получают наискорейший градиентный спуск. Оба эти метода при достаточно общих условиях сходятся к локальному минимуму со скоростью геометрической прогрессии, т. е. линейно. Если в общей схеме $\alpha_k = 1$ и $A(x_k) = F''(x_k)^{-1}$, где $F''(x_k)$ — матрица Гессе

$$F''(x_k) = \left\{ \frac{\partial^2 F}{\partial x_i \partial x_j} \right\},$$

получают метод Ньютона — Рафсона, который сходится из достаточно малой окрестности точки минимума с квадратичной скоростью. В целях увеличения области сходимости этого метода производится регулярировка шагового множителя α_k , например, способом Армийо — Голдстейна [72, 74]. Недостатком здесь является то, что при отсутствии положительной определенности матрицы Гессе вырабатываемое направление может не быть направлением убывания функции $F(x)$. Это приводит к существенному замедлению сходимости. В данном случае оказываются полезными некоторые способы приведения матрицы Гессе к положительно определенному виду (подробнее см. в работах [114, 152]).

И, наконец, при $A(x_k) = \beta_k E + F''(x_k)^{-1}$ из общей схемы получают метод Левенберга — Маркуарта, который позволяет при определенной регулярировке последовательностей $\{\alpha_k\}$ и $\{\beta_k\}$ получить квад-

¹⁸ Функция $F(x)$, определенная на выпуклом множестве D , называется сильно выпуклой, если существует такое $\gamma > 0$, при котором выполняется неравенство $F(\lambda x + (1 - \lambda)y) \leq \lambda F(x) + (1 - \lambda)F(y) - \gamma \|x - y\|^2$, $\forall x, y \in D$, $\forall \lambda \in (0, 1)$. Если это неравенство выполняется только для $\gamma = 0$, то функцию $F(x)$ называют выпуклой. Если функция выпукла в множестве D , то она и непрерывна в нем.

ратичную сходимость при более слабых требованиях к начальному приближению, чем в методе Ньютона — Рафсона.

Типичными представителями двухшаговых градиентных методов являются методы сопряженных градиентов, например, метод Флетчера — Ривза, реализуемый по схеме

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k, \\ \mathbf{p}_k &= -F'(\mathbf{x}_k) + \beta_k \mathbf{p}_{k-1}, \quad \mathbf{p}_0 = -F'(\mathbf{x}_0), \\ \beta_k &= \begin{cases} 0, & \text{если } k \text{ кратно } n, \\ \frac{\|F'(\mathbf{x}_k)\|^2}{\|F'(\mathbf{x}_{k-1})\|^2} & \text{— в противном случае,} \end{cases} \\ \alpha_k : F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) &= \min_{\alpha_k > 0} F(\mathbf{x}_k + \alpha \mathbf{p}_k), \\ \beta_k &= \frac{\|F'(\mathbf{x}_k)\|^2}{\|F'(\mathbf{x}_{k-1})\|^2}, \\ \alpha_k^* F(\mathbf{x}_k + \alpha_k^* \Phi_k) &= \min_{\alpha_k > 0} F(\mathbf{x}_k + \alpha_k \Phi_k). \end{aligned}$$

Метод сопряженных градиентов Поляка — Полака — Рибьера отличается от приведенного выше тем, что в нем

$$\beta_k = \begin{cases} 0, & \text{если } k \text{ кратно } n, \\ \frac{(F'(\mathbf{x}_k) - F'(\mathbf{x}_{k-1}))^T F'(\mathbf{x}_k)}{\|F'(\mathbf{x}_{k-1})\|^2} & \text{— в противном случае.} \end{cases}$$

Хотя эти два метода в случае квадратичной функции $F(\mathbf{x})$ теоретически эквивалентны, все же последний является в неквадратичном случае численно более устойчивым (см. [74]). Методы сопряженных градиентов обладают n -шаговой квадратичной скоростью сходимости, т. е.

$$\|\mathbf{x}_{k+n} - \mathbf{x}^*\| \leq C \|\mathbf{x}_k - \mathbf{x}^*\|^2.$$

Достоинством этих методов также является то, что для их реализации требуется малый объем оперативной памяти ЭВМ (порядка n).

Среди методов безусловной минимизации, в которых используются только градиенты минимизируемой функции, наиболее эффективны (см. [112, 144, 152]) квазиньютоновские методы (или методы переменной метрики). Их можно разделить на два класса. В одном итерации ведутся по формуле

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k J_k^{-1} F'(\mathbf{x}_k),$$

где $(n \times n)$ -матрица J_k аппроксимирует матрицу Гессе $F''(\mathbf{x}_k)$. В другом

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k H_k F'(\mathbf{x}_k),$$

где матрица H_k с размерами $n \times n$ аппроксимирует матрицу $F''(\mathbf{x}_k)^{-1}$. Матрицы J_k и H_k пересчитываются по квазиньютоновским формулам (см. п. 4 параграфа V.3). Шаговый множитель α_k выбирается, например, по правилу Армийо — Голдстейна, что существенно увеличивает область сходимости, сохраняя вблизи от точки минимума сверхлинейную скорость сходимости. На практике чаще всего используют квазиньютоновский метод Бroyдена — Флетчера — Гольдфарба — Шенно.

Методы минимизации выпуклых не обязательно дифференцируемых функций (градиент $F(\mathbf{x})$ может иметь разрывы) условно можно разбить на две группы. К одной группе относятся методы градиентного типа: различные модификации метода обобщенных градиентов, а также схемы с ускоренной сходимостью, основанные на растяжении пространства в направлении градиента или разности двух последовательных градиентов. К другой — методы «секущих плоскостей», например, метод Келли.

Из рассмотренных методов минимизации следует отметить метод оврагов для минимизации функции с сильно вытянутыми поверхностями уровня; методы покоординатного поиска с изменяемой системой координат; методы случайного поиска; комбинированные методы быстрого спуска и случайного поиска; методы дифференциального спуска; методы стохастической аппроксимации и т. д. Методам безусловной минимизации посвящена многочисленная литература (см., например, [11, 62, 63, 72, 74, 75, 80, 112, 118, 144, 152]).

Очень важными являются задачи отыскания экстремумов функции на некотором множестве. Раздел математики, посвященный изучению таких экстремумов и разработке методов нахождения экстремумов, получил название математического программирования.

Под общей задачей математического программирования понимают задачу отыскания экстремума функции $F_0(\mathbf{x})$ при условиях

$$\begin{aligned} F_j(\mathbf{x}) &= 0, \quad j = 1, 2, \dots, l, \\ F_j(\mathbf{x}) &\leq 0, \quad j = l+1, l+2, \dots, m, \quad \mathbf{x} \in Q, \end{aligned}$$

где Q — некоторое множество в пространстве векторов $\mathbf{x}$ и $1 \leq l \leq m$. Это пространство может быть как конечномерным, так и бесконечномерным. Функция $F_0(\mathbf{x})$ называется целевой, а множество

$$\begin{aligned} \Omega &= \{\mathbf{x} \in Q : F_j(\mathbf{x}) = 0, \quad j = 1, 2, \dots, l, \\ &F_j(\mathbf{x}) \leq 0, \quad j = l+1, l+2, \dots, m\} \end{aligned}$$

допустимым множеством.

Задача математического программирования принципиально отличается от классической задачи отыскания условного экстремума тем, что в ней имеются ограничения в виде равенств и неравенств. Поэтому для исследования задач математического программирования созданы свои теория и методы. Исследование свойств экстремума — главное в математическом программировании. Свойства экстремума и численные методы решения задач математического программирования определяются свойствами задач, которые в свою очередь зависят от свойств функций и множества.

Различные разделы математического программирования посвящены рассмотрению частных случаев функции $F_j(\mathbf{x})$, $j = 0, 1, \dots, m$, и различных множеств, на которых разыскивается экстремум.

К разделу линейного программирования относятся задачи математического программирования в случае, когда $\mathbf{x} \in R^n$, все функции $F_j(\mathbf{x})$ линейны, а множество Q состоит из векторов с неотрицательными

компонентами. т. е. задачи отыскания экстремума функции

$$z^* = (c, x^*) = \max (z = (c, x) : Ax \leq b, x \geq 0),$$

где $c \in E_n$, $b \in E_m$, (c, x) — скалярное произведение c и x , а матрица A имеет m строк и n столбцов.

Для решения задач линейного программирования можно использовать симплекс-метод, так называемый двойственный симплекс-метод и метод для одновременного решения прямой и двойственной задач линейного программирования (см., например, [23, 38, 120]).

Большое место в теории линейного программирования занимают конкретные задачи, среди которых особенно важны — транспортные. Для решения транспортных задач созданы специальные вычислительные методы, учитывающие структуру их ограничений (см., например, [19]).

Особое место в линейном программировании занимают задачи целочисленного программирования, в которых на допустимый вектор накладываются дополнительные требования целочисленности всех или части его компонент. Требование целочисленности компонент оптимального вектора вытекает из физического смысла многих прикладных задач. Иногда структура матрицы A такова, что при решении задачи каким-либо общим методом линейного программирования удается получить целочисленное решение, но для большинства задач линейного программирования получение целочисленного решения невозможно без процедуры поиска. Важным классом задач целочисленного программирования являются задачи, в которых часть или все переменные принимают лишь два значения: 0 либо 1. К задачам целочисленного программирования такого типа сводятся весьма сложные комбинаторные задачи о коммивояжере, задачи теории расписаний, размещения производства, раскраски графа и многие другие. Для указанных задач целочисленного программирования используются алгоритмы, основанные на методе упорядоченного перебора, методе ветвей и границ и др. (см., например, [45]).

Квадратичное программирование изучает задачи математического программирования, в которых $F_0(x) = \frac{1}{2}(xBx) + (c, x)$, где B — неположительно (неотрицательно) определенная квадратная матрица, $x, c \in R^n$, а функции $F_i(x)$ линейны. Для решения задач квадратичного программирования применимы все методы, пригодные для решения общей задачи выпуклого программирования. (О методах решения задач квадратичного программирования см., например, в [33].)

В случае, когда функция $F_0(x)$ выпукла, все функции $F_i(x)$ выпуклы и выпукло множество Q , общая задача математического программирования становится задачей выпуклого программирования. Основная особенность задачи выпуклого программирования состоит в том, что ее локальный экстремум совпадает с глобальным, т. е. наблюдается одноэкстремальность. Наиболее широкое распространение для решения задач выпуклого программирования получил метод возможных направлений. Он является обобщением классического метода наиско-

рейшего спуска в случае минимизации функций при наличии ограничений. Кроме того, для решения задач выпуклого программирования разработан ряд эффективных алгоритмов, таких как метод обобщенных градиентов, метод отсекающих гиперплоскостей, метод эллипсоидов и др. (см., например, [33, 80]).

Если хотя бы одна из функций $F_0(x)$, $F_1(x)$, ..., $F_m(x)$ нелинейна и множество Q произвольно, то основная задача математического программирования называется задачей нелинейного программирования. Для таких задач характерно наличие локальных экстремумов. Для отыскания локального экстремума задачи нелинейного программирования могут быть использованы методы выпуклого программирования. Большинство методов нелинейного программирования (см. [11, 31, 33, 74, 75, 80, 114, 112, 152]) сводят исходную задачу либо к решению системы нелинейных уравнений или последовательности задач безусловной минимизации (например, методы штрафов, методы модифицированных функций Лагранжа), либо к последовательности задач линейного или квадратичного программирования (например, метод линеаризации). Частным случаем задачи нелинейного программирования является задача геометрического программирования. В этом случае функции $F_0(x)$, $F_1(x)$, ..., $F_m(x)$ представляются в виде суммы с положительными коэффициентами произведений степенных функций переменных $x_1, x_2, \dots, x_n$, а множество Q состоит из точек с неотрицательными компонентами. Задача геометрического программирования не имеет локальных экстремумов, и для ее решения пригодны методы выпуклого программирования (см., например, [119]).

Раздел математического программирования, изучающий модели выбора оптимальных решений в условиях риска или неопределенности, получил название стохастического программирования. Существует ряд приемов сведения задач стохастического программирования к детерминированным задачам математического программирования (см., например, [23, 32]).

Для решения специальных многоэкстремальных задач используются методы динамического программирования (см., например, [8]).

Отметим, что классические задачи вариационного исчисления являются первыми примерами экстремальных задач в бесконечномерных пространствах, а классические уравнения Эйлера — первыми необходимыми условиями минимума функционалов в бесконечномерном пространстве. В настоящее время рассматриваются неклассические задачи вариационного исчисления, к которым приводят встречающиеся часто на практике задачи оптимального управления. Задачи оптимального управления отличаются от классических задач вариационного исчисления тем, что управление объекта может выбираться не на всем пространстве, а на некотором множестве, называемом множеством допустимых управлений. Необходимые условия, которым должны удовлетворять оптимальные управления, формулируются в виде принципа максимума Понтрягина (см., например, [11, 63, 74]).

В работах [31, 63, 74, 106] имеются численные методы решения задач оптимального управления, в частности методы, сводящие исходную задачу к задаче нелинейного программирования.

Наряду с созданием и теоретическим обоснованием алгоритмов решения задач проводятся работы по оценкам качества получаемых на ЭВМ решений.

В ряде работ при вычислении корней полиномов и функций используется интервальная арифметика (см., например, [157, 183]).

Одним из средств, позволяющих оценить качество вычислений, является арифметика осмысленных цифр (significant digit arithmetic-SDA). В этой арифметике каждое машинное слово, кроме порядка мантиссы, имеет метки, характеризующие надежные цифры. Разработана техника арифметических действий с такими машинными числами. Много внимания уделено задаче вычисления полиномов и их корней. В программах, осуществляющих алгоритмы решения задач, метки могут учитываться так, чтобы автоматически влиять на ход алгоритма в сторону повышения его надежности (см., например, [169]).

Учитывая изложенное выше, отметим, что в настоящее время исследования в области нелинейных уравнений сосредотачиваются вокруг следующих проблем: создания методов решения систем нелинейных уравнений большой размерности; построения методов решения задач, не зависящих от выбора начального приближения; создания методов, обладающих достаточно высокой скоростью сходимости при снижении требований к гладкости рассматриваемых функций; оценок достоверности получаемых решений; создания пакетов программ по решению систем нелинейных уравнений.

VI.1. ИНТЕРПОЛИРОВАНИЕ И СРЕДНЕКВАДРАТИЧЕСКОЕ ПРИБЛИЖЕНИЕ ФУНКЦИЙ

1. Основные понятия о приближении функций. Функция является фундаментальным понятием математики и используется в формулировках большинства задач. При решении задач на ЭВМ функция обычно представляется таблицей ее значений. Пусть, например, в области $\bar{D} = \{0 \leq x_\alpha \leq l_\alpha, \alpha = 1, 2\}$, $\bar{D} = D \cup \Gamma$, где Γ — граница (рис. 17), требуется найти решение уравнения

$$Lu = - \left[\frac{\partial}{\partial x_1} \left(k \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(k \frac{\partial u}{\partial x_2} \right) \right] = f(x_1, x_2), \quad (x_1, x_2) \in D, \quad (\text{VI.1})$$

$$k(x_1, x_2) = \begin{cases} k_1(x_1, x_2), & (x_1, x_2) \in D_1, \\ k_2(x_1, x_2), & (x_1, x_2) \in D_2, \end{cases}$$

$$D_1 = \{0 < x_1 < l_1, 0 < x_2 \leq l_2'\},$$

$$D_2 = \{0 < x_1 < l_1, l_2' \leq x_2 < l_2\},$$

которое на границе Γ должно удовлетворять условиям

$$\begin{aligned} u(x_1, 0) &= g_1(x_1), \quad \frac{\partial u}{\partial x_1}(0, x_2) = q_1(x_2), \\ u(x_1, l_2) &= g_2(x_1), \quad \frac{\partial u}{\partial x_1}(l_1, x_2) = q_2(x_2) \end{aligned} \quad (\text{VI.2})$$

и на линии разреза C условиям

$$u(x_1, l_2' + 0) - u(x_1, l_2' - 0) = r q, \quad l_1' \leq x_1 \leq l_1,$$

где

$$q = k_1 \frac{\partial u}{\partial x_2}(x_1, l_2' - 0) = k_2 \frac{\partial u}{\partial x_2}(x_1, l_2' + 0), \quad l_1' \leq x_1 \leq l_1, \quad (\text{VI.3})$$

r — некоторая постоянная. При $x_2 = l_2'$, $0 < x_1 < l_1'$ задаются условия непрерывности

$$\begin{aligned} u(x_1, l_2' + 0) &= u(x_1, l_2' - 0), \\ k_2 \frac{\partial u}{\partial x_2}(x_1, l_2' + 0) &= k_1 \frac{\partial u}{\partial x_2}(x_1, l_2' - 0), \quad 0 < x_1 < l_1'. \end{aligned} \quad (\text{VI.4})$$

Поставленная задача описывает распределение температуры в области, изображенной на рис. 17.

Решение задачи (VI.1) — (VI.4) методом сеток приводит к системе линейных алгебраических уравнений

$$Ay = b \quad (VI.5)$$

относительно таблицы искомых приближений y к функции u в точках, определяемых узлами сетки.

После получения решения системы (VI.5) может возникнуть потребность в восстановлении аналитического выражения функции по таблице ее значений или в вычислении значений функции для промежуточных значений аргумента.

Рассмотрим функцию $F(x)$, заданную на отрезке $[0, 1]$, и соответствующие ей значения $x_k, f_k, k = 0, 1, 2, \dots$. Качество этого соответствия будем оценивать по точности, с которой по значениям x_k, f_k можно восстановить значение $F(x)$ в любой точке x . Очевидно, эта точность зависит от близости значений f_k к $F(x_k)$ и плотности расположения узлов x_k .

Заметим, что отклонение f_k от $F(x_k)$ устанавливает предел точности воспроизведения, который, не исправляя таблицы, превзойти нельзя. Идеализируя задачу, будем считать, что

$$f_k = F(x_k), \quad k = 0, 1, 2, \dots \quad (VI.6)$$

В этом случае точность восстановления $F(x)$ будет определяться тем, насколько f_k хорошо описывает детали поведения функции $F(x)$, гладкостью функции $F(x)$ и, конечно, способом восстановления.

Пусть, например, способ восстановления $F(x)$ состоит в том, что строится кусочно-линейная функция, проходящая через точки (x_k, f_k) и (x_{k+1}, f_{k+1}) . В этом случае имеется линейная интерполяция вида (рис. 18)

$$P_1(x) = f_k + \frac{f_{k+1} - f_k}{x_{k+1} - x_k} (x - x_k), \quad x_k \leq x \leq x_{k+1}. \quad (VI.7)$$

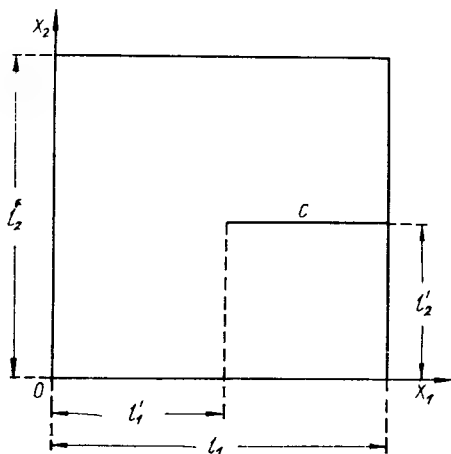


Рис. 17

Можно проводить интерполяцию с помощью квадратных, кубических и других полиномов, используя для их

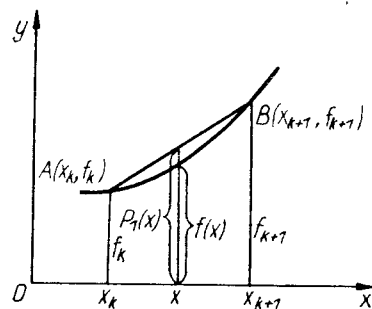


Рис. 18

построения тройки, четверки и другие точки таблицы. Рассмотрим общий случай.

Имеются $n + 1$ различных точек $x_0, x_1, \dots, x_n$ — узлов интерполяции, которым соответствуют значения $f_0, f_1, \dots, f_n$. Построим полином n -го порядка

$$P_n(x) = \sum_{m=0}^n a_m x^m, \quad (VI.8)$$

принимая в точках $x_i, i = 0, 1, \dots, n$, значения f_i . Полагая в (VI.8) $x = x_i$, получаем систему $n + 1$ уравнений для определения $n + 1$ неизвестных — коэффициентов полинома

$$\sum_{m=0}^n a_m x_i^m = f_i, \quad i = 0, 1, \dots, n. \quad (VI.9)$$

Определитель этой системы равен $\prod_{i,j} (x_i - x_j)$, т. е. отличен от нуля, так как все x_i различны. Значит, система (VI.9) имеет единственное решение — набор a_m .

Таким образом, существует единственный полином порядка не выше n , принимающий в $(n + 1)$ точке заданные значения. Он называется интерполяционным полиномом. Конкретная форма записи его может быть различна и будет рассмотрена ниже.

Оценим точность, с которой интерполяционный полином воспроизводит дифференцируемую функцию $F(x)$. Так как разность $P_n(x) - F(x)$ в узлах интерполяции $x_0, x_1, \dots, x_n$ обращается в нуль, то частное от деления этой разности на функцию

$$q(x) = (x - x_0)(x - x_1) \dots (x - x_n)$$

есть ограниченная функция, и поэтому можно написать

$$F(x) = P_n(x) + R(x)q(x).$$

Предполагая, что функция $F^{(n+1)}(x)$ существует и непрерывна, оценим $R(x)$. Для этого рассмотрим вспомогательную функцию

$$v(\xi) = F(\xi) - P_n(\xi) - R(\xi)q(\xi) = (R(\xi) - R(x))q(\xi). \quad (VI.10)$$

Очевидно, функция $v(\xi)$ обращается в нуль, по крайней мере, в $(n + 2)$ точках $x_0, x_1, \dots, x_n, x$. Следовательно, найдется хотя бы одна точка $\xi(x)$, где обращается в нуль $(n + 1)$ -я производная от $v(\xi)$, если эта производная существует и непрерывна. Дифференцируя (VI.10) $(n + 1)$ раз и подставляя $\xi = \xi(x)$, получаем

$$0 = F^{(n+1)}(\xi(x)) - R(x)(n + 1)!,$$

так как $(n + 1)$ -я производная от полинома n -й степени есть нуль, а $q(\xi)$ — полином вида $\xi^{n+1} \dots$

Таким образом,

$$R(x) = \frac{F^{(n+1)}(\xi(x))}{(n + 1)!},$$

выражение для погрешности интерполяции имеет вид

$$F(x) - P_n(x) = \frac{1}{(n+1)!} F^{(n+1)}(\xi(x)) q(x), \quad (\text{VI.11})$$

зависимость $\xi(x)$ остается неопределенной.

Если расстояние между узлами не превосходит некоторого h , т. е. $x_{k+1} - x_k \leq h$, $k = 0, 1, \dots, n-1$, то $q(x) \sim h^{n+1}$, ($h \rightarrow 0$) и из (VI.11) следует

$$|F(x) - P_n(x)| \leq c_n \max_x |F^{(n+1)}(x)| h^{n+1}, \quad (\text{VI.12})$$

где c_n — некоторая постоянная, зависящая от n . Отсюда заключаем, что при $h \rightarrow 0$ погрешность убывает, как h^{n+1} .

На практике интерполяционные формулы со сколько-нибудь большим n употребляются крайне редко. Как видно из (VI.12), увеличение степени полинома может привести к уменьшению погрешности интерполяции лишь для очень гладких функций, имеющих достаточно большое число производных. Но такой суммарной информацией о свойствах $F(x)$, как правило, не располагают. Кроме того, значения f_k вследствие погрешности округления являются всегда приближенными значениями для $F(x_k)$, поэтому полиномы, построенные по f_k и $F(x_k)$, в лучшем случае будут отличаться друг от друга на величину порядка $f_k - F(x_k)$. К тому же ошибки, содержащиеся в f , могут носить случайный характер, а это можно интерпретировать, как сильную негладкость представляемой ими функции.

Основой для применения полиномов в качестве приближений к непрерывным функциям является свойство полноты полиномов в классе непрерывных функций, выражаемое теоремой Вейерштрасса:

Если функция $f(x)$ является непрерывной на конечном замкнутом промежутке $[a, b]$, то для любого $\varepsilon > 0$ можно указать такой полином $P(x)$, что

$$|f(x) - P(x)| < \varepsilon$$

для всех x из $[a, b]$.

Важное свойство полиномов состоит в том, что они являются функциями простой природы. Чтобы вычислить значение полинома, нужно выполнить конечное число арифметических операций сложений, вычитаний и умножений, что легко реализуется на ЭВМ.

Однако из теоремы Вейерштрасса, утверждающей принципиальную возможность приближения полиномами вообще, вовсе не следует, что такое приближение можно осуществить интерполяционными полиномами, проходящими через заданные узлы, например равноудаленные. Так, если интерполировать функцию

$$f(x) = \frac{1}{1 + 25x^2}$$

на отрезке $[-1, 1]$, то интерполяционные полиномы выше шестой степени в точках, отличных от узлов интерполяции, могут сильно отличаться от значения функции. При $|x| > 0,726$ вплоть до концов сегмента появляются сильные осцилляции интерполяционного полинома

между узлами интерполяции. При этом чем выше степень полинома, тем размах осцилляций больше (подробнее об этом см. в работе [25]). Этот пример показывает, что применять интерполяцию следует весьма осторожно, изучив предварительно равномерность ее проведения. Кроме того, процедура построения интерполяционного полинома при большом количестве узлов является неустойчивой, и поэтому интерполирование следует применять локально, выбирая небольшое количество наиболее подходящих узлов.

До сих пор при восстановлении функции $F(x)$ по таблице значений x_k, f_k использовали полином $P_n(x)$ вида (VI. 8). Для этой цели можно также использовать обобщенные полиномы

$$\Phi_n(x) = \sum_{m=0}^n a_m \varphi_m(x) \quad (\text{VI.13})$$

различных порядков. В (VI.13) $\varphi_m(x)$, $m = 0, 1, \dots, n$, — известные (заданные) функции, которые могут задаваться кусочно. При использовании обобщенных полиномов можно произвести соответствующее исследование возможности и качества аппроксимации.

В настоящее время интенсивно развивается раздел теории приближения функций, получивший название сплайн-интерполяций.

Пусть на отрезке $a \leq x \leq b$ в точках $a = x_0 < x_1 < \dots < x_N = b$ задаются соответствующие ординаты $y_0, y_1, \dots, y_N$. Функция $S(x)$, непрерывная на $[a, b]$ вместе со своими производными до $(r-1)$ -й включительно, совпадающая с полиномом r -й степени на отрезке $x_{j-1} \leq x \leq x_j$, $j = 1, 2, \dots, n$, и удовлетворяющая условиям

$$S(x_j) = y_j, \quad j = 0, 1, \dots, N,$$

называется сплайном r -й степени.

Широкое применение нашли кубические сплайны, обладающие следующим свойством минимальной нормы: среди всех функций $f(x)$, имеющих на $[a, b]$ непрерывную вторую производную, и таких, что $f(x_i) = y_i$, $i = 0, 1, \dots, N$, кубический сплайн, для которого $S''(a) = S''(b) = 0$, минимизирует интеграл

$$\|f\|^2 = \int_a^b |f''(x)|^2 dx.$$

Благодаря последнему свойству сплайн-аппроксимация имеет ряд преимуществ по сравнению с другими процессами-аппроксимациями.

Сплайн непрерывен и имеет непрерывные первую и вторую производные, а третья производная может претерпевать в точках, где стыкуются два полинома, разрыв с конечным скачком.

Близкая к проблеме интерполяции задача возникает и в том случае, когда значения заданной функции $\varphi(x)$ известны в узловых точках x_k не точно, а с некоторой погрешностью, максимальное значение которой для каждой точки задается в качестве априорной информации. В этом случае задача состоит в построении такой кривой, которая в известном смысле наилучшим образом аппроксимировала бы функцию, заданную со случайными погрешностями в узловых точках. Такую

задачу обычно решают, минимизируя выражение

$$\sum_{k=0}^n (P_m(x_k) - f_k)^2 = \delta \quad (\text{VI.14})$$

или

$$\sum_{k=0}^n (\Phi_m(x) - f_k)^2 = \delta \quad (\text{VI.15})$$

по параметрам a_i , $i = 0, 1, \dots, m$. Этот способ приближений получил название метода наименьших квадратов.

2. Интерполяционный полином Лагранжа. Использование в качестве интерполяционного полинома, записанного в (VI.8), вследствие громоздкости выражения коэффициентов a_m через x_i , f_i нецелесообразно. В зависимости от условий задачи интерполяции и заданной априорной информации о свойствах интерполируемой функции применяются различные интерполяционные полиномы.

Пусть функция $y = f(x)$ в точках $x_0, x_1, \dots, x_n$ принимает соответственно значения $y_0, y_1, \dots, y_n$. Выпишем полином степени, не превосходящей n , который в данных $(n+1)$ точках $x_0, x_1, \dots, x_n$ принимал значения $y_0, y_1, \dots, y_n$:

$$\begin{aligned} P_n(x) = & \frac{y_0(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)} + \frac{y_1(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)} + \dots \\ & \dots + \frac{y_n(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})}, \end{aligned} \quad (\text{VI.16})$$

который называется интерполяционным полиномом Лагранжа. Непосредственной проверкой убеждаемся, что этот полином совпадает с y_0 при $x = x_0$, с y_1 при $x = x_1$ и т. д. В силу утверждения в п. 1 настоящего параграфа этот полином является единственным, удовлетворяющим условиям задачи.

Погрешность интерполяции определяется выражением

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-x_0)\dots(x-x_n),$$

где $\xi \in [\min x_i; \max x_i]$, $i = 0, 1, \dots, n$.

Интерполяционный полином Лагранжа для неравных промежутков изменения аргумента применяют, когда используют имеющиеся экспериментальные данные или когда неудобно или невозможно получить таблицу равноотстоящих значений аргумента, а также при обратном интерполировании (об этом см. п. 10 настоящего параграфа).

Пример 1. Для функции $y = \sin \pi x$ построить интерполяционный полином Лагранжа, выбрав узлы интерполирования

$$x_0 = 0, \quad x_1 = \frac{1}{6}, \quad x_2 = \frac{1}{2}.$$

Решение. Вычисляем соответствующие значения функции

$$y_0 = 0, \quad y_1 = \sin \frac{\pi}{6} = \frac{1}{2}, \quad y_2 = \sin \frac{\pi}{2} = 1.$$

Применяя формулу (VI.16), получаем

$$P_2(x) = \frac{\left(x - \frac{1}{6}\right)\left(x - \frac{1}{2}\right)}{-\frac{1}{6}\left(-\frac{1}{2}\right)} \cdot 0 + \frac{x\left(x - \frac{1}{2}\right)}{\frac{1}{6}\left(\frac{1}{6} - \frac{1}{2}\right)} \cdot \frac{1}{2} + \frac{x\left(x - \frac{1}{6}\right)}{\frac{1}{2}\left(\frac{1}{2} - \frac{1}{6}\right)} \cdot 1,$$

или

$$P_2(x) = \frac{7}{2}x - 3x^2.$$

На вычисление значения функции $f(x)$ в одной точке по формуле Лагранжа требуется $(2n^2 + 3n)$ операций сложения, $(2n^2 + n - 1)$ операций умножения, $(n+1)$ операций деления.

При использовании интерполяционного полинома Лагранжа следует иметь в виду, что точность решения для крайних отрезков падает. Существенным недостатком интерполяционной формулы Лагранжа является то, что построение интерполяционного полинома Лагранжа по его значениям в точках $x_0, x_1, \dots, x_{n+1}$ совершенно не использует полином Лагранжа, построенный по точкам $x_0, x_1, \dots, x_n$. Таким образом, добавление одной точки x_{n+1} приводит к тому, что всю работу по построению полинома Лагранжа приходится проводить заново.

3. Полином Чебышева и выбор узлов интерполирования. В некоторых задачах имеется возможность выбрать узлы интерполирования в интересах уменьшения погрешности произвольно. Возникает естественная необходимость задания узлов интерполирования на отрезке $[a, b]$ таким образом, чтобы верхний предел,

$$|R_n(x)| \leq \frac{M_{n+1}[a, b] \max_{a \leq x \leq b} |q(x)|}{(n+1)!}, \quad (\text{VI.17})$$

$$M_{n+1} = \max_{a \leq x \leq b} |f^{(n+1)}(x)|, \quad q(x) = (x-x_0)\dots(x-x_n),$$

модуля погрешности интерполяции (VI.11) был наименьшим.

Оказывается, что если в качестве узлов интерполяции взять нули полинома Чебышева, то погрешность интерполирования для данной функции будет наименьшей.

Полиномы Чебышева определяются равенством

$$T_m(x) = \cos(m \arccos x), \quad |x| \leq 1, \quad m = 0, 1, 2, \dots,$$

при $m = 0$

$$T_0(x) = \cos 0 = 1,$$

при $m = 1$

$$T_1(x) = \cos(\arccos x) = x,$$

при $m = 2$

$$T_2(x) = \cos(2 \arccos x) = 2 \cos^2(\arccos x) - 1 = 2x^2 - 1.$$

Далее, с учетом тождества

$$\cos(m+1)\theta = 2 \cos \theta \cos m\theta - \cos(m-1)\theta$$

при $\theta = \arccos x$ получаем следующую рекуррентную формулу:

$$T_{m+1}(x) = 2xT_m(x) - T_{m-1}(x). \quad (\text{VI.18})$$

Применяя ее, последовательно находим

$$T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1,$$

$$T_5(x) = 16x^5 - 20x^3 + 5x.$$

.....

Соотношение (VI.18) показывает, что все $T_m(x)$, $m = 0, 1, 2, \dots$, действительно, являются полиномами; степень полинома $T_m(x)$ равняется m и коэффициент при старшей степени x равен 2^{m-1} .

Полином $T_m(x)$ имеет ровно m вещественных корней, которые определяем следующим образом: из

$$T_m(x) = \cos(m \arccos x) = 0$$

находим

$$m \arccos x = \frac{\pi}{2}(2k+1), \quad \text{или} \quad x = \cos \frac{(2k+1)\pi}{2m}.$$

При $k = 0, 1, \dots, m-1$ получаем m различных корней, причем все они лежат между -1 и $+1$.

Из определения $T_m(x)$ следует, что $\max_{-1 \leq x \leq 1} |T_m(x)|$ не превышает 1. Решая уравнения $\cos(m \arccos x) = \pm 1$, находим, что этот максимум равен 1 и достигается в точках

$$\hat{x}_k = \cos \frac{k\pi}{m}, \quad k = 0, 1, \dots, m.$$

Для полинома

$$q_m(x) = (x - x_0)(x - x_1) \dots (x - x_{m-1}),$$

где $x_0, x_1, \dots, x_{m-1}$ — корни полинома $T_m(x)$, имеем соотношения

$$q_m(x) = \frac{1}{2^{m-1}} T_m(x), \quad \max_{-1 \leq x \leq 1} |q_m(x)| = \frac{1}{2^{m-1}}.$$

И этот максимум достигается в точках $\hat{x}_k = \cos \frac{k\pi}{m}$, $k = 0, 1, \dots, m$. Используя это, можно показать [3, 71], что какой бы полином $Q(x)$ степени m с коэффициентом при x^m , равным 1, мы не взяли, $\max |Q(x)| \geq \frac{1}{2^{m-1}}$, и, следовательно, полином $q_m(x) = \frac{1}{2^{m-1}} T_m(x)$ является полиномом, имеющим на отрезке $[-1, 1]$ наименьший максимум. Это свойство полиномов Чебышева позволяет ответить на вопрос о выборе узлов интерполирования. Если интерполирование производится на отрезке $[-1, 1]$, то значение $\max_{-1 \leq x \leq 1} |q_{n+1}(x)|$, входящее в оценку (VI.17), будет наименьшим возможным значением при условии, что в качестве узлов интерполирования $x_0, x_1, \dots, x_n$ взяты

корни полинома Чебышева $T_{n+1}(x)$. В этом случае равномерная оценка погрешности интерполирования (VI.17) принимает следующий вид:

$$|R_n(x)| \leq M_{n+1}(-1, 1) \frac{1}{(n+1)!} \frac{1}{2^n},$$

где

$$M_{n+1}(-1, 1) = \max_{-1 \leq x \leq 1} f^{(n+1)}(x).$$

Если интерполирование производится на произвольном отрезке $[a, b]$, то линейной заменой

$$u = \frac{2}{b-a}x + \frac{a+b}{a-b} \quad (a \leq x \leq b)$$

отрезок $[a, b]$ можно перевести в отрезок $[-1, 1]$.

Обратная замена будет

$$x = \frac{1}{2}[(b-a)u + (b+a)] \quad (-1 \leq u \leq 1),$$

следовательно, при такой замене корни полинома $T_{n+1}(u)$ $u_k = \cos \frac{2k+1}{2(n+1)}\pi$ переходят в

$$x_k = \frac{1}{2} \left[(b-a) \cos \frac{2k+1}{2(n+1)}\pi + a+b \right], \quad k = 0, 1, \dots, n.$$

Равномерная оценка (VI.17) для этого случая будет такова:

$$|R_n(x)| \leq \frac{M_{n+1}(a, b)(b-a)^{n+1}}{(n+1)! 2^{n+1}}, \quad M_{n+1}(a, b) = \max_{a \leq x \leq b} f^{(n+1)}(x),$$

причем эта оценка на рассматриваемом отрезке является наилучшей [3].

Для вычисления чебышевских узлов x_k , $k = 0, 1, \dots, n$, требуется произвести $2n + (n+1)k + 4$ операций сложения, $3n + (n+1) \times \times l + 3$ операций умножения и $1 + (n+1)q$ операций деления, где k, l и q соответственно количества операций сложения, умножения и деления, которые затрачиваются при вычислении функции $y = \cos x$ в одной точке.

4. Конечные разности и конечные разделенные разности. При рассмотрении п. 2 параграфа VI.1 интерполяционного метода Лагранжа подразумевается, что множество используемых узлов известно. Однако часто известным является лишь требуемая точность, а множество узлов, которые следует использовать, определяется информацией о том, как вычисляется функция. Вполне понятна важность такого метода, который давал бы возможность строить интерполяционный полином, если число используемых узлов увеличено, без повторения всего вычисления. Справиться с этой задачей позволяет теория конечных разностей.

Нисходящей конечной разностью первого порядка функции $y = f(x)$ с шагом h называется разность $f(x+h) - f(x)$. Ее будем обозначать символом $\Delta f(x)$ (или Δy):

$$\Delta f(x) = \Delta y = f(x+h) - f(x).$$

Символ Δ обладает ассоциативным свойством:

$$\Delta(f_1(x) + f_2(x)) = \Delta f_1(x) + \Delta f_2(x).$$

Разность $\Delta f(x)$ является некоторой новой функцией аргумента x , и для нее мы можем рассматривать разность $\Delta(\Delta f(x))$. Тогда по отношению к функции $f(x)$ имеем разность второго порядка

$$\Delta^2 y = \Delta^2 f(x) = \Delta(\Delta f(x)) = f(x + 2h) - 2f(x + h) + f(x).$$

Конечные разности высших порядков определяются из низших совершенно аналогично.

Предположим, что функция $y = f(x)$ на отрезке $[a, b]$ задана следующим образом: для значений $x = x_0, x_0 + h, x_0 + 2h, \dots, x_0 + nh$ известны значения функции $y = f(x)$:

$$f(x_0) = y_0, \quad f(x_1) = y_1, \quad \dots, \quad f(x_0 + nh) = y_n.$$

Интервал между двумя последовательными значениями аргумента x равен h и, следовательно, первую разность функции $y = f(x)$ в точке $x = x_i$ получим из выражения

$$\Delta y_i = y_{i+1} - y_i.$$

Разность второго порядка $\Delta^2 y_i$ получим, если возьмем первую разность от Δy_i , т. е.

$$\Delta^2 y_i = \Delta(\Delta y_i) = \Delta y_{i+1} - \Delta y_i.$$

Аналогично имеем

$$\Delta^3 y_i = \Delta(\Delta^2 y_i) = \Delta^2 y_{i+1} - \Delta^2 y_i,$$

$$\dots$$

$$\Delta^k y_i = \Delta(\Delta^{k-1} y_i) = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i.$$

Эти разности удобно вычислять, построив табл. 10, в которой разности в любом столбце получаем вычитанием двух соседних разностей в столбце слева. Табл. 10 можем продолжить, подключая последовательно новые узлы.

Рассмотрим теперь таблицу значений $f(x_i)$ в точках $x_i, i = 0, 1, \dots, n$, которые расположены на отрезке $[a, b]$ неравномерно. Для каждых двух последовательных узлов x_i и x_{i+1} образуем отношение

$$\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}.$$

Его называют первой разделенной разностью и обозначают символом $f(x_i; x_{i+1})$ (или $[x_i x_{i+1}]$).

Вторая разделенная разность определяется следующим образом:

$$f(x_i; x_{i+1}; x_{i+2}) = [x_i x_{i+1} x_{i+2}] = \frac{f(x_{i+1}; x_{i+2}) - f(x_i; x_{i+1})}{x_{i+2} - x_i}.$$

Разделенные разности высших порядков определяются аналогично, причем на каждом шаге знаменатель представляет собой разность тех

Таблица 10

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y \dots$
x_0	y_0	Δy_0	$\Delta^2 y_0$	$\Delta^3 y_0$	$\Delta^4 y_0$	$\Delta^5 y_0$
$x_0 + h$	y_1	Δy_1	$\Delta^2 y_1$	$\Delta^3 y_1$	$\Delta^4 y_1$	$\Delta^5 y_1$
$x_0 + 2h$	y_2	Δy_2	$\Delta^2 y_2$	$\Delta^3 y_2$	$\Delta^4 y_2$	$\Delta^5 y_2$
$x_0 + 3h$	y_3	Δy_3	$\Delta^2 y_3$	$\Delta^3 y_3$	$\Delta^4 y_3$	$\Delta^5 y_3$
$x_0 + 4h$	y_4	Δy_4	$\Delta^2 y_4$	$\Delta^3 y_4$	$\Delta^4 y_4$	$\Delta^5 y_4$
$x_0 + 5h$	y_5	Δy_5	$\Delta^2 y_5$	$\Delta^3 y_5$	$\Delta^4 y_5$	$\Delta^5 y_5$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Таблица 11

x	f	$f(x)$	$f(x; x)$	$f(x; x; x)$	$\dots$
x_0	$f(x_0)$	$f(x_0; x_1)$	$f(x_0; x_1; x_2)$	$f(x_0; x_1; x_2; x_3)$	$\dots$
x_1	$f(x_1)$	$f(x_1; x_2)$	$f(x_1; x_2; x_3)$	$f(x_1; x_2; x_3; x_4)$	$\dots$
x_2	$f(x_2)$	$f(x_2; x_3)$	$f(x_2; x_3; x_4)$	$f(x_2; x_3; x_4; x_5)$	$\dots$
x_3	$f(x_3)$	$f(x_3; x_4)$	$f(x_3; x_4; x_5)$	$f(x_3; x_4; x_5; x_6)$	$\dots$
x_4	$f(x_4)$	$f(x_4; x_5)$	$f(x_4; x_5; x_6)$	$f(x_4; x_5; x_6; x_7)$	$\dots$
x_5	$f(x_5)$	$f(x_5; x_6)$	$f(x_5; x_6; x_7)$	$f(x_5; x_6; x_7; x_8)$	$\dots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

значений x , которые только один раз встречаются в числителе:

$$f(x_i; x_{i+1}; \dots; x_{i+k}) = \frac{f(x_{i+1}; x_{i+2}; \dots; x_{i+k}) - f(x_i; x_{i+1}; \dots; x_{i+k-1})}{x_{i+k} - x_i}.$$

Разделенные разности обладают следующим свойством: для функции $f(x) = \varphi(x) + \psi(x)$ имеем

$$f(x_i; x_{i+1}; \dots; x_{i+k}) = \varphi(x_i; x_{i+1}; \dots; x_{i+k}) + \psi(x_i; x_{i+1}; \dots; x_{i+k}). \quad (\text{VI.19})$$

При вычислении разделенных разностей составляем табл. 11. Многоточия указывают на то, что таблицу можно продолжить, подключая последовательно новые узлы интерполирования.

Будем также рассматривать разделенные разности, которые содержат переменную x . По определению

$$\begin{aligned} f(x; x_0) &= \frac{f(x_0) - f(x)}{x_0 - x}, \\ f(x; x_0; x_1) &= \frac{f(x_0; x_1) - f(x, x_0)}{x_1 - x}, \\ &\dots \\ f(x; x_0; x_1; \dots, x_n) &= \frac{f(x_0; x_1; x_2; \dots; x_n) - f(x; x_0; x_1; \dots; x_{n-1})}{x_n - x}. \end{aligned} \quad (\text{VI.20})$$

Рассмотрим, например, какой вид имеют такие разности для функции $f(x) = x^r$, где r — некоторое натуральное число. Имеем

$$f(x, x_0) = \frac{x_0^r - x^r}{x_0 - x} = x_0^{r-1} + x_0^{r-2}x + \dots + x^{r-1},$$

т. е. первая разделенная разность представляет собой полином степени $r - 1$. В силу соотношения (VI.19) это свойство сразу же устанавливается и для любого полинома степени r . Разделенная разность порядка r полинома степени r будет равняться некоторой постоянной.

5. Интерполяционная формула Ньютона. Из равенств (VI.20), определяющих разделенные разности и содержащих переменную x , находим

$$\begin{aligned} f(x) &= f(x_0) + f(x; x_0)(x - x_0), \\ f_1(x; x_0) &= f(x_0; x_1) + f(x; x_0; x_1)(x - x_1), \\ f_2(x; x_0; x_1) &= f(x_0; x_1; x_2) + f(x; x_0; x_1; x_2)(x - x_2), \\ &\dots \\ f_{n+1}(x; x_0; x_1; \dots; x_n) &= f_n(x_0; x_1; \dots; x_{n-1}) + \\ &\quad + f_{n+1}(x; x_0; x_1; \dots; x_n)(x - x_n). \end{aligned} \quad (\text{VI.21})$$

Подставляя в первое равенство (VI.21) разделенную разность $f_1(x; x_0)$ из второго равенства (VI.21), находим

$$f(x) = f(x_0) + (x - x_0)f(x_0; x_1) + (x - x_0)(x - x_1)f(x; x_0; x_1). \quad (\text{VI.22})$$

Далее, подставляя в полученное равенство вместо $f_2(x; x_0; x_1)$ его выражение из (VI.21), находим

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0)f(x_0; x_1) + (x - x_0)(x - x_1)f(x_0; x_1; x_2) + \\ &\quad + (x - x_0)(x - x_1)(x - x_2)f(x; x_0; x_1; x_2). \end{aligned} \quad (\text{VI.23})$$

Этот процесс можем продолжить. Замечая закономерность образования правых частей получаемых равенств и применяя метод полной математической индукции, окончательно находим

$$f(x) = \hat{P}_n(x) + \hat{R}_n(x), \quad (\text{VI.24})$$

где

$$\begin{aligned} \hat{P}_n(x) &= f(x_0) + f(x_0; x_1)(x - x_0) + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots \\ &\quad \dots + f_n(x_0; x_1; x_2; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned} \quad (\text{VI.25})$$

есть полином степени n и

$$\hat{R}_n(x) = f(x; x_0; x_1; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_n). \quad (\text{VI.26})$$

Значения функции $f(x)$ совпадают со значениями интерполяционного полинома Лагранжа $P_n(x)$ в точках $x = x_0, x_1, \dots, x_n$ (см. п. 2 параграфа IV.1), поэтому разделенные разности для функции $P_n(x)$, взятые в этих $n + 1$ точках, совпадают с соответствующими разностями для $f(x)$. Следовательно, из формулы (VI.24) получаем

$$P_n(x) = \hat{P}_n(x) + P_n(x; x_0; x_1; \dots; x_n)(x - x_0) \dots (x - x_n), \quad (\text{VI.27})$$

где $\hat{P}_n(x)$ — полином, представленный формулой (VI.25). Разделенная разность $P_n(x; x_0; x_1; \dots; x_n)$, входящая в равенство (VI.27), представляет собой $(n + 1)$ -ю разделенную разность полинома $P_n(x)$ степени n . Поэтому $P_n(x; x_0; \dots; x_n) = 0$ и

$$P_n(x) = \hat{P}_n(x); \quad \hat{R}_n(x) = R_n(x)$$

для всех значений x .

Формулу Ньютона запишем теперь с остаточным членом в форме Коши:

$$f(x) = P_n(x) + \frac{f^{(n+1)}(\xi) q(x)}{(n+1)!}, \quad (\text{VI.28})$$

где

$$\begin{aligned} P_n(x) &= f(x_0) + f(x_0; x_1)(x - x_0) + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots \\ &\quad \dots + f_n(x_0; x_1; x_2; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}), \\ q(x) &= (x - x_0)(x - x_1) \dots (x - x_n), \quad \xi \in (\alpha, \beta), \end{aligned} \quad (\text{VI.29})$$

$$\alpha = \min\{x; x_0, x_1, \dots, x_n\}, \quad \beta = \max\{x, x_0, x_1, \dots, x_n\}.$$

Полином $P_n(x)$, вычисляемый по формуле (VI.29), называется интерполяционным полиномом Ньютона. В отличие от полинома Лагранжа прибавление новой пары значений (x_{n+1}, y_{n+1}) сводится здесь просто к прибавлению одного нового члена.

Пример 2. Составить таблицу разделенных разностей и построить интерполяционный полином Ньютона по следующим данным:

x	0	2	3	5
$f(x)$	1	3	2	5

Решение. Таблица 11 в этом случае принимает вид

Таблица 12

x	f	$f(\cdot)$	$f(\cdot\cdot)$	$f(\cdot\cdot\cdot)$
0	1			
2	3	1	$-\frac{2}{3}$	$\frac{3}{10}$
3	2	-1	$\frac{5}{6}$	
5	5	$\frac{3}{2}$		

Применяя формулу (VI.29) и используя значения из табл. 12, получаем

$$P_3(x) = 1 + 1 \cdot x + \left(-\frac{2}{3}\right)x(x-2) + \frac{3}{10}x(x-2)(x-3).$$

Раскрывая скобки, имеем

$$P_3(x) = \frac{3}{10}x^3 - \frac{13}{6}x^2 + \frac{62}{15}x + 1.$$

На вычисление значения функции $f(x)$ в одной точке x по формуле Ньютона требуются $n^2 + 2n$ операций сложения, $\frac{n^2 + n}{2}$ операций умножения и $\frac{n(n+1)}{2}$ операций деления.

При равноотстоящих узлах вычисление разделенных разностей можно заменить вычислением простых разностей. Для этого применяются интерполяционные формулы Грегори — Ньютона для интерполирования вперед или назад и различные формулы центральных разностей, такие, например, как формулы Гаусса, Стирлинга и др. [5, 6].

6. Формула Грегори — Ньютона для интерполирования вперед. Пусть функция $f(x)$ задана таблицей ее значений в узлах

$$x_0, x_1 = x_0 + h, x_2 = x_0 + 2h, \dots, x_n = x_0 + nh.$$

Рассмотрим, какой вид будет иметь в этом частном случае таблица конечных разделенных разностей (табл. 13).

Продолжая таблицу, устанавливаем, что в этом случае по диагонали расположены следующие разности:

$$y_0, \frac{1}{h} \Delta y_0, \frac{1}{2!h^2} \Delta^2 y_0, \dots, \frac{1}{n!h^n} \Delta^n y_0.$$

Следовательно, формула Ньютона (VI.29) принимает в этом частном случае вид

$$f(x) = P_n(x) + R_n(x), \quad (\text{VI.30})$$

где

$$P_n(x) = y_0 + \frac{\Delta y_0}{h}(x-x_0) + \frac{\Delta^2 y_0}{2!h^2}(x-x_0)(x-x_1) + \dots$$

Таблица 13

x	f	$f(\cdot)$	$f(\cdot\cdot)$	$f(\cdot\cdot\cdot)$	$f(\cdot\cdot\cdot\cdot)$
x_0	y_0				
		$\frac{1}{h} \Delta y_0$			
$x_1 = x_0 + h$	y_1		$\frac{1}{2!h^2} \Delta^2 y_0$		
		$\frac{1}{h} \Delta y_1$		$\frac{1}{3!h^3} \Delta^3 y_0$	
$x_2 = x_0 + 2h$	y_2		$\frac{1}{2!h^2} \Delta^2 y_1$		$\frac{1}{4!h^4} \Delta^4 y_0$
		$\frac{1}{h} \Delta y_2$		$\frac{1}{3!h^3} \Delta^3 y_1$	
$x_3 = x_0 + 3h$	y_3		$\frac{1}{2!h^2} \Delta^2 y_2$		
		$\frac{1}{h} \Delta y_3$			
$x_4 = x_0 + 4h$	y_4				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$$\dots + \frac{\Delta^n y_0}{n!h^n}(x-x_0)(x-x_1)\dots(x-x_{n-1}),$$

$$R_n(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_n)}{(n+1)!} f^{(n+1)}(\xi), \quad \xi \in (\alpha, \beta),$$

$$\alpha = \min\{x, x_0, \dots, x_n\}, \quad \beta = \max\{x, x_0, \dots, x_n\}.$$

Вместо переменной x введем новую переменную t , полагая $x = x_0 + th$. При этом $x - x_k = (t - k)h$ и формула (VI.30) принимает вид

$$f(x) = P_n(x) + R_n(x), \quad (\text{VI.31})$$

где

$$\begin{aligned} P_n(x) &= P_n(x_0 + th) = \\ &= y_0 + t\Delta y_0 + \frac{t(t-1)}{2!} \Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!} \Delta^3 y_0 + \dots \\ &\dots + \frac{t(t-1)(t-2)\dots(t-n+1)}{n!} \Delta^n y_0, \end{aligned} \quad (\text{VI.32})$$

$$R_n(x) = R_n(x_0 + th) = \frac{t(t-1)\dots(t-n)}{(n+1)!} h^{n+1} f^{(n+1)}(\xi), \quad t = \frac{x-x_0}{h}.$$

Это есть формула Грегори — Ньютона для интерполирования вперед.

Обычно при практических вычислениях интерполяционная формула Грегори — Ньютона обрывается на членах, содержащих такие разности, которые в пределах заданной точности можно считать постоянными.

Предполагая, что разности $\Delta^{n+1}y$ почти постоянны для функции $y = f(x)$ и h достаточно мало и, учитывая, что

$$f^{(n+1)}(x) = \lim_{h \rightarrow 0} \frac{\Delta^{n+1}y}{h^{n+1}},$$

приближенно можно положить

$$f^{(n+1)}(\xi) \simeq \frac{\Delta^{n+1}y_0}{h^{n+1}}.$$

В этом случае остаточный член формулы Грегори — Ньютона

$$R_n(x) \simeq \frac{t(t-1) \dots (t-n)}{(n+1)!} \Delta^{n+1}y_0.$$

Формула Грегори — Ньютона (VI.32) обычно применяется для интерполирования в точках, близких к левому концу отрезка.

Пример 3. Пусть имеем таблично заданную функцию:

x	0,1	0,2	0,3	0,4	0,5
y	0,09983	0,19867	0,29552	0,38942	0,47943.

используя формулу Грегори — Ньютона интерполирования вперед, необходимо вычислить $\sin 0,14$.

Решение. Строим таблицу разностей

Таблица 14

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
0,1	0,09983	9884			
0,2	0,19867		-199		
0,3	0,29552	9685	-295	-96	
0,4	0,38942	9390	-389	-94	2
0,5	0,47943	9001			

Заметим, что в столбцах разностей не указаны десятичные разряды (они ясны из столбца значений функции). Применяя разности, лежащие на нисходящей диагонали, строим интерполяционный полином вида (VI.32)

$$P_4(x) = 0,09983 + 0,09884t - 0,00199 \frac{t(t-1)}{2!} - 0,00096 \frac{t(t-1)(t-2)}{3!} + 0,00002 \frac{t(t-1)(t-2)(t-3)}{4!}, \quad t = \frac{x - x_0}{h}.$$

При $t = \frac{0,14 - 0,1}{0,1} = 0,4$ имеем

$$\sin 0,14 \simeq 0,09983 + 0,09884 \cdot 0,4 - 0,00199 \cdot \frac{0,4 \cdot (-0,6)}{2} - 0,00096 \frac{0,4 \cdot (-0,6) \cdot (-1,6)}{2} + 0,00002 \frac{0,4 \cdot (-0,6) \cdot (-1,6) \cdot (-2,6)}{24} \simeq 0,13954.$$

На вычисление значения функции $f(x)$ в одной точке x по формуле Грегори — Ньютона требуются $n - 1$ операций деления, $\left(\frac{n^2 + 3n}{2} - 1\right)$ операций умножения и $\frac{n^2 + 3n}{2}$ операций сложения.

7. Формула Грегори — Ньютона для интерполирования назад. В формуле Грегори — Ньютона для интерполирования назад используются разности, стоящие в табл. 13 на восходящей диагонали, начинающейся с y_n . В этом случае имеем

$$f(x) = P_n(x) + R_n(x), \quad (VI.33)$$

где

$$P_n(x) = P_n(x_n + th) = y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!} \Delta y_{n-2} + \frac{t(t+1)(t+2)}{3!} \Delta^3 y_{n-1} + \dots + \frac{t(t+1) \dots (t+n-1)}{n!} \Delta^n y_0,$$

$$R_n(x) = R_n(x_n + th) = \frac{t(t+1) \dots (t+n)}{(n+1)!} h^{n+1} f^{(n+1)}(\xi), \quad \xi \in (\alpha, \beta),$$

$$\alpha = \min \{x, x_0, \dots, x_n\}, \quad \beta = \max \{x, x_0, \dots, x_n\},$$

$$t = \frac{x - x_n}{h}.$$

Формулу (VI.33) обычно применяют для интерполирования в правом конце отрезка. В предположении, что разности $\Delta^{n+1}y$ почти постоянны для функции $y = f(x)$ и h достаточно мало, имеем

$$R_n(x) \simeq \frac{t(t+1) \dots (t+n)}{(n+1)!} \Delta^{n+1}y_n.$$

Иногда для записи формулы Грегори — Ньютона интерполирования назад используют другое обозначение разностей. Для узловых точек $x_i = x_0 + ih$, $i = 0, 1, \dots, n$, и их соответствующих ординат $y_i = f(x_i)$ под восходящими разностями будем понимать разности, определяемые следующими равенствами

$$\nabla y_i = f(x_i) - f(x_{i-1}) = y_i - y_{i-1},$$

$$\nabla^2 y_i = \nabla(\nabla y_i) = \nabla y_i - \nabla y_{i-1},$$

$$\nabla^3 y_i = \nabla(\nabla^2 y_i) = \nabla^2 y_i - \nabla^2 y_{i-1},$$

$$\dots \dots \dots$$

Нетрудно убедиться, что

$$\nabla^k y_i = \Delta^k y_{i-k},$$

и формулу Грегори — Ньютона (VI.33) можно переписать так:

$$f(x) = P_n(x) + R_n(x), \quad (VI.34)$$

где

$$P_n(x) = P_n(x_n + th) =$$

$$= y_n + t \nabla y_n + \frac{t(t+1)}{2!} \nabla^2 y_n + \frac{t(t+1)(t+2)}{3!} \nabla^3 y_n + \dots$$

$$\dots + \frac{t(t+1)(t+2) \dots (t+n-1)}{n!} \nabla^n y_n,$$

$$R_n(x) = R_n(x_n + th) = \frac{t(t+1) \dots (t+n)}{(n+1)!} h^{n+1} f^{(n+1)}(\xi), \quad \xi \in (\alpha, \beta).$$

Пример 4. Используя табл. 14, вычислить $\sin 0,46$.

Решение. Так как значение аргумента $x = 0,46$ лежит вблизи конечного узла интерполирования, для вычисления $\sin 0,46$ используем формулу Грегори — Ньютона интерполирования назад. Имеем

$$t = \frac{0,46 - 0,5}{0,1} = -0,4.$$

Применяя разности, лежащие на восходящей диагонали табл. 14, по формуле (VI.34) находим

$$\sin 0,46 \approx 0,479\,43 + (-0,43) \cdot 0,090\,01 - \frac{(-0,4) \cdot (0,6)}{2} 0,003\,89 -$$

$$- \frac{(-0,4) \cdot 0,6 \cdot 0,6}{6} 0,000\,94 + \frac{(-0,4) \cdot 0,6 \cdot 0,6 \cdot 2,6}{24} 0,000\,02 = 0,443\,95.$$

Число арифметических операций на вычисление значения функции $f(x)$ в одной точке такое же, как и в формуле Грегори — Ньютона интерполирования вперед.

8. Формулы интерполирования с центральными разностями. Пусть значения функции $y = f(x)$ заданы в точках

$$x = x_0 - nh, \quad x_0 - (n-1)h, \dots, x_0 - h, x_0, x_0 + h, \dots$$

$$\dots, x_0 + (n-1)h, x_0 + nh,$$

тогда разности задаются табл. 15.

Таблица 15

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{-3}	y_{-3}						
		Δy_{-3}					
x_{-2}	y_{-2}		$\Delta^2 y_{-3}$				
		Δy_{-2}		$\Delta^3 y_{-3}$			
x_{-1}	y_{-1}		$\Delta^2 y_{-2}$		$\Delta^4 y_{-3}$		
		$\left[\begin{smallmatrix} \Delta y_{-1} \\ \Delta y_0 \end{smallmatrix} \right]$	$\Delta^2 y_{-1}$	$\left[\begin{smallmatrix} \Delta^3 y_{-2} \\ \Delta^3 y_{-1} \end{smallmatrix} \right]$	$\Delta^4 y_{-2}$	$\left[\begin{smallmatrix} \Delta^5 y_{-3} \\ \Delta^5 y_{-2} \end{smallmatrix} \right]$	$\Delta^6 y_{-3} \dots$
x_0	y_0		$\Delta^2 y_0$		$\Delta^4 y_{-1}$		
x_1	y_1	Δy_1		$\Delta^3 y_0$			
x_2	y_2	Δy_2	$\Delta^2 y_1$				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Применяя интерполяционную формулу Ньютона с разделенными разностями $f(x_0)$, $f(x_0; x_1)$, $f(x_0; x_1; x_{-1})$, $f(x_0; x_1; x_{-1}, x_2)$, ..., $f(x_0; x_1; \dots; x_n; x_{-n})$ и полагая $x = x_0 + th$, находим следующую формулу:

$$f(x) = P_{2n}(x) + R_{2n}(x), \quad (\text{VI.35})$$

где

$$P_{2n}(x) = P_{2n}(x_0 + th) =$$

$$= y_0 + t \Delta y_0 + \frac{t(t-1)}{2!} \Delta^2 y_{-1} + \frac{t(t-1)(t+1)}{3!} \Delta^3 y_{-1} +$$

$$+ \frac{t(t-1)(t+1)(t-2)}{4!} \Delta^4 y_{-2} + \frac{t(t-1)(t+1)(t-2)(t+2)}{5!} \Delta^5 y_{-2} + \dots$$

$$\dots + \frac{t(t-1)(t+1) \dots (t-n+1)(t+n-1)}{(2n-1)!} \Delta^{2n-1} y_{-(n-1)} +$$

$$+ \frac{t(t-1)(t+1) \dots (t+n-1)(t-n)}{(2n)!} \Delta^{2n} y_{-n},$$

$$R_{2n}(x) = R_{2n}(x_0 + th) = \frac{t(t^2-1)(t^2-2^2) \dots (t^2-n^2)}{(2n+1)!} h^{2n+1} f^{(2n+1)}(\xi),$$

$$\xi \in (x_{-n}, x_n).$$

Это есть первая интерполяционная формула Гаусса, в которой используются разности, расположенные в табл. 15 вдоль нижней ломаной.

Применяя интерполяционную формулу Ньютона с разделенными разностями $f(x_0)$, $f(x_0; x_{-1})$, $f(x_0; x_{-1}; x_1)$, ..., $f(x_0; x_{-1}; \dots; x_{-n}; x_n)$, получаем вторую формулу Гаусса

$$f(x) = P_{2n}(x) + R_{2n}(x), \quad (\text{VI.36})$$

где

$$P_{2n}(x) = P_{2n}(x_0 + th) =$$

$$= y_0 + t \Delta y_{-1} + \frac{t(t+1)}{2!} \Delta^2 y_{-1} + \frac{t(t+1)(t-1)}{3!} \Delta^3 y_{-2} +$$

$$+ \frac{t(t+1)(t-1)(t+2)}{4!} \Delta^4 y_{-2} + \frac{t(t+1)(t-1)(t+2)(t-2)}{5!} \Delta^5 y_{-3} + \dots$$

$$\dots + \frac{t(t+1)(t-1) \dots (t+n-1)(t-n+1)}{(2n-1)!} \Delta^{2n-1} y_{-n} +$$

$$+ \frac{t(t+1)(t-1) \dots (t+n)}{(2n)!} \Delta^{2n} y_{-n},$$

а остаточный член имеет такое же выражение, как и в первой формуле Гаусса.

В формуле (VI.36) используются центральные разности, расположенные в таблице вдоль верхней ломаной. Интерполяционные полиномы первой и второй формул Гаусса совпадают.

Использование разностей, расположенных вдоль одной из ломаных, делает формулы (VI.35), (VI.36) несколько неудобными для употребления. Чаще применяется формула Стирлинга, которую получаем, взяв среднее арифметическое первой и второй интерполяционных

формулы Гаусса:

$$f(x) = P_{2n}(x) + R_{2n}(x), \quad (\text{VI.37})$$

где

$$\begin{aligned} P_{2n}(x) = & y_0 + t \frac{\Delta y_{-1} + \Delta y_0}{2} + \frac{t^2}{2!} \Delta^2 y_{-1} + \frac{t(t^2-1)}{3!} \frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} + \\ & + \frac{t^2(t^2-1)}{4!} \Delta^4 y_{-2} + \frac{t(t^2-1)(t^2-2^2)}{5!} \frac{\Delta^5 y_{-3} + \Delta^5 y_{-2}}{2} + \dots \\ & \dots + \frac{t(t^2-1)(t^2-2^2) \dots [t^2-(n-1)^2]}{(2n-1)!} \frac{\Delta^{2n-1} y_{-n} + \Delta^{2n-1} y_{-n+1}}{2} + \\ & + \frac{t^2(t^2-1)(t^2-2^2) \dots [t^2-(n-1)^2]}{(2n)!} h^{2n+1} f^{(2n+1)}(\xi), \quad \xi \in (x_{-n}, x_n), \\ & t = \frac{x - x_0}{h}. \end{aligned}$$

Если аналитическое выражение функции $f(x)$ неизвестно, то при малом h полагают

$$R_{2n}(x) \simeq \frac{t(t^2-1)(t^2-2^2) \dots (t^2-n^2)}{(2n+1)!} \frac{\Delta^{2n+1} y_{-n-1} + \Delta^{2n+1} y_{-n}}{2}.$$

Рассматриваемые выше интерполяционные формулы будут иметь симметричную форму, если для записи разностей ввести новый символ δ (символ центральных разностей).

Первая центральная разность $\delta f(x)$ функции $f(x)$ определяется формулой

$$\delta f(x) = f\left(x + \frac{h}{2}\right) - f\left(x - \frac{h}{2}\right). \quad (\text{VI.38})$$

Если принять, что $y_t = f(x_t)$, где $x_t = x_0 + th$ и t принимает любое вещественное значение, то равенство (VI.38) можно записать в виде

$$\delta y_t = y_{t+\frac{1}{2}} - y_{t-\frac{1}{2}}.$$

Рассматривая разность $\delta f(x)$ как некоторую новую функцию переменной x , вторую центральную разность функции $f(x)$ определяем следующим образом:

$$\delta^2 f(x) = \delta(\delta f(x)) = \delta f\left(x + \frac{h}{2}\right) - \delta f\left(x - \frac{h}{2}\right),$$

или

$$\delta^2 y_t = \delta y_{t+\frac{1}{2}} - \delta y_{t-\frac{1}{2}}.$$

Центральную разность порядка k определяем аналогичным образом:

$$\delta^k f(x) = \delta^{k-1} f\left(x + \frac{h}{2}\right) - \delta^{k-1} f\left(x - \frac{h}{2}\right),$$

Таблица 16

x	y	δy	$\delta^2 y$	$\delta^3 y$	$\delta^4 y$	$\delta^5 y$	$\delta^6 y$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{-3}	y_{-3}						
		δy					
		$-\frac{5}{2}$					
x_{-2}	y_{-2}		$\delta^2 y_{-2}$				
		δy		$\delta^3 y$			
		$-\frac{3}{2}$		$-\frac{3}{2}$			
x_{-1}	y_{-1}		$\delta^2 y_{-1}$		$\delta^4 y_{-1}$		
		$\left[\begin{array}{c} \delta y \\ \delta y \end{array} \right]$		$\left[\begin{array}{c} \delta^3 y \\ \delta^3 y \end{array} \right]$		$\left[\begin{array}{c} \delta^5 y \\ \delta^5 y \end{array} \right]$	
x_0	y_0	$\left[\begin{array}{c} \delta y \\ \delta y \end{array} \right]$	$\delta^2 y_0$	$\left[\begin{array}{c} \delta^3 y \\ \delta^3 y \end{array} \right]$	$\delta^4 y_0$	$\left[\begin{array}{c} \delta^5 y \\ \delta^5 y \end{array} \right]$	$\delta^6 y_0 \dots$
x_1	y_1		$\delta^2 y_1$		$\delta^4 y_1$		
		δy		$\delta^3 y$			
		$\frac{3}{2}$		$\frac{3}{2}$			
x_2	y_2		$\delta^2 y_2$				
		δy					
		$\frac{5}{2}$					
x_3	y_3						
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

или

$$\delta^k y_t = \delta^{k-1} y_{t+\frac{1}{2}} - \delta^{k-1} y_{t-\frac{1}{2}}.$$

Центральные разности запишем в виде табл. 16.

Эта таблица отличается от табл. 15 только записью, используемой для обозначения разностей. Фактические величины разностей, занимающих в этих таблицах одинаковые места, те же.

Введем оператор μ , обозначающий среднее значение двух соседних чисел в каждом столбце табл. 15:

$$\mu \delta y_0 = \frac{1}{2} (\delta y_{-\frac{1}{2}} + \delta y_{\frac{1}{2}}), \quad \mu \delta^3 y_{\frac{1}{2}} = \frac{1}{2} (\delta^3 y_0 + \delta^3 y_1),$$

$$\mu \delta^3 y_0 = \frac{1}{2} (\delta y_{-\frac{1}{2}} + \delta y_{\frac{1}{2}}) \text{ и т. д.}$$

В новых обозначениях формула Стирлинга (VI.37) принимает вид

$$f(x) = P_{2n}(x) + R_{2n}(x),$$

где

$$\begin{aligned} P_{2n}(x) = & y_0 + t \mu \delta y_0 + \frac{t^2}{2!} \delta^2 y_0 + \frac{t(t^2-1)}{3!} \mu \delta^3 y_0 + \frac{t^2(t^2-1)}{4!} \delta^4 y_0 + \dots \\ & \dots + \frac{t(t^2-1)(t^2-2^2) \dots [t^2-(n-1)^2]}{(2n-1)!} \mu \delta^{2n-1} y_0 + \end{aligned}$$

$$+ \frac{t^2(t^2-1)(t^2-2^2) \dots [t^2-(n-1)^2]}{(2n)!} \delta^{2n} y_0, \quad (\text{VI.39})$$

$$R_{2n}(x) = \frac{t(t^2-1)(t^2-2^2) \dots (t^2-n^2)}{(2n+1)!} h^{2n+1} f^{(2n+1)}(\xi), \quad \xi \in (x_{-n}, x_n),$$

$$t = \frac{x - x_0}{h}.$$

Формула Стирлинга обычно применяется для интерполирования в точках, близких к середине отрезка.

Пример 5. Используя таблицу разностей (табл. 17)

Т а б л и ц а 17

x	$y = \sin x$	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
10°	0,17365	16 837			
20°	0,34202		—1039		
30°	0,50000	15 788		—480	
			—1519		45
40°	0,64279	14 279	—1954	—435	
50°	0,76604	12 325			

необходимо найти $\sin 32^\circ$.

Решение. Значение $x = 32^\circ$ лежит вблизи значения $x_0 = 30^\circ$, которое будем считать центральным узлом. Выделяем в табл. 17 нужные разности и записываем формулу Стирлинга

$$\sin x \approx 0,5 + \frac{0,15798 + 0,14279}{2} t - 0,01519 \frac{t^2}{2} - \frac{0,00480 + 0,00435 t(t^2-1)}{3!} + 0,00045 \frac{t^2(t^2-1)}{4!}.$$

Отсюда при $t = \frac{32^\circ - 30^\circ}{10^\circ} = 0,2$ находим

$$\sin 32^\circ \approx 0,52992.$$

На вычисление значения функции $f(x)$ в одной точке по формуле Стирлинга требуются $2n^2 + 5n - 1$ операций сложения, $6n + 1$ операций умножения, $2n - 1$ операций деления.

9. **Экстраполирование.** В пунктах 1 и 2 параграфа IV.1 приведены примеры применения интерполяционных формул Грегори — Ньютона для отыскания значений функции, соответствующих промежуточным значениям аргумента, отсутствующим в таблице. Эти же формулы могут быть использованы и для нахождения значений функции, соответствующих значениям аргумента, находящимся вне пределов таблицы, т. е. для экстраполирования. Применение для этой цели интерполяционных формул и составленных программ ничем не отличается от их применения для интерполирования в узком смысле.

Если $x^* < x_0$ и x^* близко к x_0 , то для вычисления $y^* = f(x^*)$ выгодно применять первую интерполяционную формулу Грегори — Ньютона [31], причем в этом случае

$$t = \frac{x^* - x_0}{h} < 0.$$

Если же $x^* > x_n$ и значение x^* близко к x_n , то для вычисления y^* удобнее использовать вторую интерполяционную формулу Грегори — Ньютона (VI.34), причем

$$t = \frac{x^* - x_n}{h} > 0.$$

Формула Стирлинга, содержащая центральные разности, используется для интерполирования в середине отрезка, и применять ее для экстраполирования нецелесообразно.

10. **Обратное интерполирование.** Пусть требуется по таблично заданной функции $y = f(x)$ отыскать значение аргумента x^* , которому соответствует данное значение y^* , отсутствующее в таблице.

Предположим, что на рассматриваемом отрезке функция монотонна и, следовательно, имеет однозначную функцию: $x = \varphi(y)$. В этом случае обратное интерполирование сводится к обычному интерполированию для функции $\varphi(y)$.

Для приближенного вычисления значения x^* применяем формулу Лагранжа (VI.16) или формулу Ньютона (VI.28).

Пример 6. По значениям таблично заданной функции

x	1	2	2,5	3
y	—6	—1	5,625	16

найти значение x^* , для которого $y^* = 0$.

Решение. Рассматривая заданные значения функции, можем предположить, что функция монотонная на всем отрезке [2, 5]. Применяя формулу Лагранжа, получаем

$$x^* \approx L(y) = 1 \frac{(y+1)(y-5,625)(y-16)}{(-5)(-11,625)(-22)} + 2 \frac{(y+6)(y-16)(y-5,625)}{5(-17)(-6,625)} +$$

$$+ 3 \frac{(y+6)(y+1)(y-5,625)}{22 \cdot 17 \cdot 10,375} + 2,5 \frac{(y+6)(y+1)(y-16)}{11,625 \cdot 6,625(-10,375)}.$$

Для $y^* = 0$ будем иметь $x^* \approx 2,122$.

Если функция $y = f(x)$ не является монотонной на отрезке $[a, b]$ и, следовательно, не имеет однозначной обратной, то этот отрезок разбиваем на отрезки монотонности заданной функции. И тогда на каждом таком отрезке обратное интерполирование производим изложенным выше способом. Другой способ обратного интерполирования функции, не имеющей однозначной обратной, состоит в следующем: строим интерполирующий функцию $f(x)$ полином $P_n(x)$ степени n . Для приближенного вычисления искомого значения x^* решаем алгебраическое уравнение

$$P_n(x) = y^*.$$

Конечно, фиксирование значения y^* еще не достаточно для начала вычислений; здесь необходимо предварительно найти промежуток, в котором может лежать искомое значение аргумента, т. е. отделить это значение, и только после этого приближаться к нему с какой угодно степенью точности.

11. **Интерполяционный полином Эрмита.** Если табулирована не только функция, но и ее производная вплоть до некоторого порядка,

то можно потребовать, чтобы в узлах интерполяции совпадали не только значения искомой функции $y = f(x)$ и интерполяционной функции, но и значения их производных вплоть до некоторого порядка. Такую интерполяцию называют Эрмитовой. Следуя работе [18], рассмотрим интерполяционный полином Эрмита.

Пусть на отрезке $[a, b]$ заданы узлы

$$x_0, x_1, \dots, x_m$$

и соответствующие им системы чисел

$$y_0, y_0^{(1)}, y_0^{(2)}, \dots, y_0^{(\gamma_0)},$$

$$y_1, y_1^{(1)}, y_1^{(2)}, \dots, y_1^{(\gamma_1)},$$

• • • • •

$$y_m, y_m^{(1)}, y_m^{(2)}, \dots, y_m^{(\gamma_m)},$$

где $\gamma_0, \dots, \gamma_m$ заданные натуральные числа.

Рассмотрим следующую задачу: построить полином $P_n(x)$ степени $n = \gamma_0 + \gamma_1 + \dots + \gamma_m + m$, который обладал бы свойствами

$$P_n^{(l)}(x_k) = y_k^{(l)}, \quad k = 0, 1, \dots, m; \quad l = 0, 1, \dots, \gamma_k.$$

Нетрудно убедиться, что такой полином единственный. Предположим, что существуют два таких полинома. Тогда их разность, которую обозначим через $Q_n(x)$, должна удовлетворять равенствам

$$Q_n^{(l)}(x_k) = 0, \quad k = 0, 1, \dots, m, \quad l = 0, 1, \dots, \gamma_k,$$

и, следовательно, точки x_k должны быть нулями полинома $Q_n(x)$ соответственно кратностей $\gamma_k + 1$. Отсюда полином $Q_n(x)$ степени n должен делиться на полином

$$\prod_{k=0}^m (x - x_k)^{\gamma_k + 1}$$

степени $n + 1$, что возможно, только если $Q_n(x) \equiv 0$. Теперь покажем, что полином, удовлетворяющий заданным условиям, существует. Легко убедиться, что такой полином может быть записан следующим образом:

$$P_n(x) = \sum_{k=0}^m \sum_{l=0}^{\nu_k} y_k^{(l)} P_{k,l}(x), \quad (\text{VI.40})$$

где

$$P_{k,l}(x) = \frac{\Omega(x)}{l!(x-x_k)^{v_k+1-l}} \left\{ \frac{(x-x_k)^{v_k+1}}{\Omega(x)} \right\}_{(x_k)^{(v_k-l)}},$$

$$\Omega(x) = \prod_{\alpha} (x-x_k)^{v_k+1}.$$

Выражение в фигурных скобках $\{F(x)\}_{(a)}^{(\lambda)}$ обозначает сумму членов разложения функции $F(x)$ в ряд Тейлора в окрестности $x = a$, в ко-

торое разность $(x - a)$ входит в степенях, не превышающих натуральное число λ .

Отметим, что полином $P_{k,l}(x)$ подобран так, чтобы он имел степень n и удовлетворял условиям

$$P_{k,l}^{(i)}(x) = \begin{cases} 1, & \text{если одновременно } k = s, l = i, \\ 0, & \text{в остальных случаях, } k = 0, 1, \dots, m; l = 0, 1, \dots, \gamma_k. \end{cases}$$

Это следует из того, что

$$\varphi(x) \left\{ \frac{1}{\varphi(x)} \right\}_{(a)}^{(u)} = 1 + k(x-a)^{u+1} + \dots, \quad (\text{VI.41})$$

поскольку μ -й отрезок разложения Тейлора функции (VI.41) совпадает с μ -м отрезком соответствующего разложения единицы:

$$1 = \varphi(x) \frac{1}{\varphi(x)}.$$

Полиному (VI.40) соответствует общая интерполяционная формула Эрмита

$$f(x) = P_n(x) + R_n(x),$$

где

$$P_n(x) = \sum_{k=0}^m \sum_{l=0}^{v_m} f^{(l)}(x_k) P_{k,l}(x).$$

Остаточный член $R_n(x)$ определяется по формуле [18]

$$R_n(x) = \frac{f^{(n+1)}(\theta)}{(n+1)!} \Omega(x), \quad \Omega(x) = \prod_{k=0}^m (x - x_k)^{\gamma_k+1}, \quad (\text{VI.42})$$

$$\theta \in [\alpha, \beta], \quad \alpha = \min \{x, x_0, x_1, \dots, x_m\},$$

$$\beta = \max \{x, x_0, x_1, \dots, x_m\}.$$

12. Интерполирование функций кубическими полиномиальными сплайнами. Пусть в узлах сетки

$$a = x_0 < x_1 < \cdots < x_k < x_{k+1} < \cdots < x_{N+1} = b$$

заданы значения y_k функции $y = f(x)$. Рассмотрим, как ставится и решается задача интерполяции кусочно-кубическими сплайнами. На отрезке $[a, b]$ необходимо найти функцию $g(x)$, удовлетворяющую следующим требованиям:

функция $g(x)$ принадлежит классу непрерывных вместе со своими производными до второго порядка включительно, т. е. $g(x) \in C^{(2)}[a, b]$;

на каждом из отрезков $[x_k, x_{k+1}]$ функция $g(x)$ является кубическим полиномом вида

$$g(x) = g(x_k) = \sum_{l=0}^3 a_l^{(k)} (x - x_k)^l, \quad k = 0, 1, \dots, N; \quad (\text{VI.43})$$

в узлах сетки $\{x_k\}$ для этой функции выполняются равенства

$$g(x_k) = y_k, \quad k = 0, 1, \dots, N+1;$$

функция $g(x)$ удовлетворяет условиям

$$\frac{d^2 g}{dx^2}(a) = \frac{d^2 g}{dx^2}(b) = 0.$$

Поставленная задача нахождения интерполирующей кусочно-кубической функции (VI.43) в предположениях, сделанных в работе [20], имеет единственное решение, и коэффициенты $a_0^{(k)}, a_1^{(k)}, a_2^{(k)}, a_3^{(k)}$, $k = 0, 1, \dots, N$, в полиноме (VI.43) находят следующим образом:

$$a_0^{(k)} = y_k, \quad a_1^{(k)} = \frac{\Delta y_k}{h_k} - \frac{s_k}{2} h_k - \frac{\Delta s_k}{6} h_k, \quad a_2^{(k)} = \frac{s_k}{2}, \quad a_3^{(k)} = \frac{\Delta s_k}{6 h_k},$$

$$k = 0, 1, \dots, N, \quad (\text{VI.44})$$

где $\Delta y_k = y_{k+1} - y_k$, $h_k = \Delta x_k = x_{k+1} - x_k \neq 0$, а вектор $\hat{s} = (s_1, s_2, \dots, s_N)^T$ является решением системы линейных алгебраических уравнений

$$C \hat{s} = B \hat{y}, \quad (\text{VI.45})$$

где $\hat{y} = (y_0, y_1, \dots, y_{N+1})^T$, матрица

$$C = \begin{bmatrix} \frac{1}{3}(h_0 + h_1) & \frac{h_1}{6} & 0 & \dots & 0 & 0 \\ \frac{h_1}{6} & \frac{1}{3}(h_1 + h_2) & \frac{h_2}{6} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \frac{h_{N-1}}{6} \frac{1}{3}(h_{N-1} + h_N) \end{bmatrix}$$

$$(\text{VI.46})$$

с размерами $N \times N$, а матрица

$$B = \begin{bmatrix} \frac{1}{h_0} - \left(\frac{1}{h_0} + \frac{1}{h_1}\right) & \frac{1}{h_1} & 0 & \dots & 0 & 0 \\ 0 & \frac{1}{h_1} & -\left(\frac{1}{h_1} + \frac{1}{h_2}\right) & \frac{1}{h_2} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{h_{N-1}} - \left(\frac{1}{h_{N-1}} + \frac{1}{h_N}\right) \frac{1}{h_N} \end{bmatrix}$$

порядка $N \times (N+2)$. При решении системы (VI.45) обычно применяют метод решения системы линейных алгебраических уравнений с учетом трехдиагональной структуры матриц. Для удовлетворения последнего из требований, накладываемых на функцию $g(x)$, полагают, что $s_0 = s_{N+1} = 0$.

Матрица (VI.46) системы линейных алгебраических уравнений (VI.45) симметрична и имеет диагональное преобладание, поэтому

для решения этой системы целесообразно использовать метод квадратных корней, учитывающих ленточную структуру матрицы.

Отметим, что функция $g(x)$ доставляет минимум функционалу

$$\mathcal{J}(u) = \int_a^b \frac{d^2 u}{dx^2} dx, \quad u(x_k) = y_k,$$

где в качестве функций сравнения берутся дважды непрерывно-дифференцируемые функции.

Оценка погрешности интерполяции $\varphi(x) = g(x) - f(x)$ удовлетворяет неравенству

$$\max_{a < x < b} |\varphi(x)| \leq Ch^\alpha,$$

где α, C — неотрицательные константы, не зависящие от сетки, и

$$h = \max_{1 \leq k \leq N+1} |x_k - x_{k-1}|.$$

Значение постоянной α можно получить достаточно точно, если иметь дополнительную информацию относительно функции $f(x)$. Так, например, если $f(x)$ имеет непрерывную первую производную, то $\alpha = 1$. Если же $f(x)$ непрерывная и имеет непрерывные производные до второго порядка включительно, то $\alpha = 2$ [2, 88].

Пример 7. Построить кубический полиномиальный сплайн, приближающий следующую таблично заданную функцию

$$\begin{array}{ccccc} x & 0 & 0,25 & 0,5 & 1 \\ y & 0 & -0,01660156 & -0,69629 & 2. \end{array}$$

Решение. В этом случае имеем

$$N = 3, \quad h_0 = 0,25 - 0 = 0,25, \quad h_1 = 0,5 - 0,25 = 0,25, \quad h_2 = 0,7 - 0,5 = 0,2, \\ h_3 = 1 - 0,7 = 0,3,$$

$$C = \begin{bmatrix} \frac{1}{3}(h_0 + h_1) & \frac{h_1}{6} & 0 \\ \frac{h_1}{6} & \frac{1}{3}(h_1 + h_2) & \frac{h_2}{6} \\ 0 & \frac{h_2}{6} & \frac{1}{3}(h_2 + h_3) \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & \frac{1}{24} & 0 \\ \frac{1}{24} & \frac{3}{20} & \frac{1}{30} \\ 0 & \frac{1}{30} & \frac{1}{6} \end{bmatrix} \approx$$

$$\approx \begin{bmatrix} 0,16666667 & 0,04166667 & 0 \\ 0,04166667 & 0,15 & 0,03333333 \\ 0 & 0,03333333 & 0,16666667 \end{bmatrix},$$

$$B = \begin{bmatrix} \frac{1}{h_0} - \left(\frac{1}{h_0} + \frac{1}{h_1}\right) & \frac{1}{h_1} & 0 & 0 \\ 0 & \frac{1}{h_1} & -\left(\frac{1}{h_1} + \frac{1}{h_2}\right) & \frac{1}{h_2} \\ 0 & 0 & \frac{1}{h_2} & -\left(\frac{1}{h_2} + \frac{1}{h_3}\right) \frac{1}{h_3} \end{bmatrix} =$$

$$= \begin{bmatrix} 4 & -8 & 4 & 0 & 0 \\ 0 & 4 & -9 & 5 & 0 \\ 0 & 0 & 5 & -8\frac{1}{3} & 3\frac{1}{3} \end{bmatrix} \approx \begin{bmatrix} 4 & -8 & 4 & 0 & 0 \\ 0 & 4 & -9 & 5 & 0 \\ 0 & 0 & 5 & -8,333\ 333\ 33 & 3,333\ 333\ 33 \end{bmatrix},$$

$$\hat{B}\hat{y} = \begin{bmatrix} 4 & -8 & 4 & 0 & 0 \\ 0 & 4 & -9 & 5 & 0 \\ 0 & 0 & 5 & -8,333\ 333\ 33 & 3,333\ 333\ 33 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -0,016\ 601\ 56 \\ -0,218\ 75 \\ -0,696\ 29 \\ -2 \end{bmatrix} \approx$$

$$\approx \begin{bmatrix} -0,742\ 187\ 5 \\ -1,578\ 906\ 25 \\ -1,958 \end{bmatrix}.$$

Находим теперь вектор $\hat{s} = (s_1, s_2, s_3)^T$, решая систему линейных уравнений $C\hat{s} = \hat{B}\hat{y}$. Имеем:

$$s_1 = -2,568\ 938\ 08, \quad s_2 = -7,536\ 747\ 665, \quad s_3 = 10,240\ 650\ 42.$$

Полагая еще $s_0 = s_4 = 0$, находим коэффициенты функции (VI.43) по формулам (VI.44):

$$\begin{aligned} a_0^{(0)} &= 0; \quad a_1^{(0)} = 0,040\ 617\ 16; \quad a_2^{(0)} = 0; \quad a_3^{(0)} = 1,712\ 374\ 61; \\ a_0^{(1)} &= 0,016\ 601\ 56; \quad a_1^{(1)} = -0,280\ 453\ 08; \quad a_2^{(1)} = -1,284\ 280\ 96; \\ &\quad a_3^{(1)} = -3,313\ 126\ 96, \\ a_0^{(2)} &= -0,218\ 75; \quad a_1^{(2)} = -1,543\ 804\ 86; \quad a_2^{(2)} = 3,769\ 126\ 18; \\ &\quad a_3^{(2)} = -2,251\ 747\ 65; \\ a_0^{(3)} &= -0,696\ 29; \quad a_1^{(3)} = -3,321\ 665\ 05; \quad a_2^{(3)} = -5,120\ 174\ 76; \\ &\quad a_3^{(3)} = 5,689\ 083\ 07. \end{aligned}$$

Подставляя в функцию (VI.43) для различных значений $k = 0, 1, \dots, N$ коэффициенты a_i^k , получаем четыре кубических полинома, аппроксимирующих на каждом из отрезков $[0; 0,25]$, $[0,25; 0,5]$, $[0,5; 0,7]$, $[0,7; 1]$ таблично заданную функцию y .

При интерполировании кубическими полиномиальными сплайнами требуется произвести $13n + 3$ сложений, $11n + 4$ умножений, $8n + 1$ делений.

13. Интерполирование функций двух переменных. Пусть в прямоугольнике $a \leq x \leq b$, $c \leq y \leq d$ заданы значения некоторой функции $f(x, y)$ в точках (x_i, y_j) , $i = 0, 1, \dots, n$, $j = 0, 1, \dots, m$, требуется найти значение функции $f(x, y)$ в некоторой промежуточной точке (x^*, y^*) . Так, ставится задача интерполирования функций двух переменных. Она более громоздкая по сравнению с соответствующей задачей для функций одной переменной. Интерполяционные полиномы, построенные для функций двух переменных, имеют сложную форму и очень неудобные для практического применения. Поэтому представляется целесообразным при решении задачи определения значения

функции $f(x, y)$ для промежуточных значений аргументов использовать метод последовательного интерполирования по каждой переменной отдельно.

Сначала находят значение $f(x^*, y_0)$, интерполируя по x , например по интерполяционной формуле Лагранжа для одной переменной. При этом используют табличные значения функции $f(x, y)$ в точках (x_0, y_0) , (x_1, y_0) , ..., (x_n, y_0) . Затем, интерполируя по x , находят $f(x^*, y_1)$, используя значения функции в точках (x_0, y_1) , (x_1, y_1) , ..., (x_n, y_1) . Аналогичным образом вычисляют $f(x^*, y_2)$, $f(x^*, y_3)$, ..., $f(x^*, y_m)$. Получив приближенные значения $f(x^*, y_j)$, $j = 0, 1, \dots, m$, находят $f(x^*, y^*)$, интерполируя по y на отрезке $[y_0, y_m]$. Вместо формулы Лагранжа можно также использовать формулу Ньютона для неравноотстоящих узлов.

Если значения функции $f(x, y)$ заданы в равноотстоящих узлах (с шагом h_1 по направлению оси x и с шагом h_2 по направлению оси y), то можно применить одну из формул Грегори — Ньютона или формулу Стирлинга. Отметим, что в результате интерполирования функции $f(x, y)$ отдельно по каждой переменной получают значение интерполирующего полинома в заданной точке, а не его аналитическое выражение.

14. Среднеквадратическое приближение таблично заданных функций с помощью полиномов. В ряде случаев значения приближаемой функции берут из экспериментов и, следовательно, эти данные имеют случайные погрешности. При этом нецелесообразно требовать, чтобы приближаемая и приближающая функции в заданных точках совпадали точно. В этих условиях для приближения функций целесообразно использовать метод наименьших квадратов или среднеквадратичное приближение функции.

Задача о среднеквадратичном приближении функций может быть поставлена следующим образом. Пусть в узлах $x_0, x_1, \dots, x_n$ имеются значения $y_0, y_1, \dots, y_n$ функции $y = f(x)$. Эти значения могут быть приближенными. Среди полиномов m -й степени ($m < n$)

$$P_m(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m \quad (\text{VI.47})$$

следует найти такой, который доставляет минимум выражению

$$\delta = \sum_{i=0}^n [P_m(x_i) - y_i]^2. \quad (\text{VI.48})$$

Неизвестными являются коэффициенты полинома (VI.47). Сумма (VI.48) представляет собой квадратичную форму от этих коэффициентов. Кроме того, формула (VI.48) показывает, что функция $\delta = \delta(a_0, a_1, \dots, a_m)$ не может принимать отрицательных значений. Следовательно, минимум функции δ существует. Применяя необходимые условия экстремума функции δ , получаем систему линейных алгебраических уравнений для определения коэффициентов $a_0, a_1, \dots, a_m$:

$$\begin{aligned} \frac{1}{2} \frac{\partial \delta}{\partial a_k} &= a_0 \sum_{i=0}^n x_i^k + a_1 \sum_{i=0}^n x_i^{k+1} + \dots \\ &\dots + a_m \sum_{i=0}^n x_i^{k+m} - \sum_{i=0}^n y_i x_i^k = 0. \end{aligned} \quad (\text{VI.49})$$

Полагая

$$c_p = \sum_{i=0}^n x_i^p, \quad d_p = \sum_{i=0}^n y_i x_i^p,$$

записываем систему (VI.49) в виде

$$Ca = d, \quad (VI.50)$$

где

$$C = \begin{bmatrix} c_0 & c_1 & \dots & c_m \\ c_1 & c_2 & \dots & c_{m+1} \\ \dots & \dots & \dots & \dots \\ c_m & c_{m+1} & \dots & c_{2m} \end{bmatrix} \quad (VI.51)$$

есть матрица системы, a — вектор неизвестных, $a = (a_0, a_1, \dots, a_m)^T$, d — вектор правых частей системы, $d = (d_0, d_1, \dots, d_m)^T$. Известно [4, 13], что если среди узлов $x_0, x_1, \dots, x_n$ нет совпадающих и $m \leq n$, то система (VI.50) имеет единственное решение и ее матрица является положительно определенной.

Пусть $a_0 = a_0^*$, $a_1 = a_1^*$, ..., $a_m = a_m^*$ — решение системы (VI.50). Тогда полином $P_m(x) = a_0^* + a_1^*x + \dots + a_m^*x^m$ является единственным полиномом степени m , обладающим минимальным квадратичным отклонением $\delta^* = \delta \min$.

Погрешность среднеквадратического приближения функции характеризуется величиной

$$\Delta = \sqrt{\frac{\sum_{i=0}^n |f(x_i) - P_m^*(x_i)|^2}{n+1}}.$$

Для решения системы линейных алгебраических уравнений (VI.50) целесообразно применять метод квадратных корней (см. п. 4 параграфа 1.3).

Заметим, что при увеличении степени аппроксимирующего полинома обусловленность матрицы (VI.51) ухудшается. В случае плохой обусловленности вместо системы (VI.50) приходится решать систему со сдвигом спектра матрицы

$$(C + \alpha E)a = d,$$

где $\alpha > 0$ — некоторый малый параметр, E — единичная матрица (см. [27]).

Пример 8. Построить методом наименьших квадратов полином второй степени по следующим данным:

$$\begin{array}{cccccc} x & -1 & -0,4 & 0 & 0,5 & 1 \\ y & -2 & -1 & 0 & 1,2 & 2,05 \end{array}$$

Решение. Имеем

$$\begin{aligned} c_0 &= 1 + 1 + 1 + 1 + 1 = 5, & c_1 &= -1 - 0,4 + 0 + 0,5 + 1 = 0,1, \\ c_2 &= 1 + 0,16 + 0 + 0,25 + 1 = 2,41, & c_3 &= -1 - 0,064 + 0 + 0,125 + 1 = 0,061, \end{aligned}$$

$$c_4 = 1 + 0,0256 + 0 + 0,0625 + 1 = 2,0881,$$

$$d_0 = -2 - 1 + 0 + 1,2 + 2,05 = 0,25,$$

$$d_1 = -1 \cdot (-2) - 0,4 \cdot (-1) + 0 + 0,5 \cdot 1,2 + 1 \cdot 2,05 = 5,05,$$

$$d_2 = 1 \cdot (-2) + 0,16 \cdot (-1) + 0 + 0,25 \cdot 1,2 + 1 \cdot 2,05 = 0,19.$$

Решая систему линейных алгебраических уравнений $Ca = d$, где

$$C = \begin{bmatrix} 5 & 0,1 & 2,41 \\ 0,1 & 2,41 & 0,061 \\ 2,41 & 0,061 & 2,0881 \end{bmatrix}, \quad d = \begin{bmatrix} 0,25 \\ 5,05 \\ 0,19 \end{bmatrix},$$

находим следующие коэффициенты искомого полинома:

$$a_0 \approx -0,01410, \quad a_1 \approx 2,09485, \quad a_2 \approx 0,04607.$$

Полином имеет вид

$$P_2(x) = -0,01410 + 2,09485x + 0,04607x^2.$$

На вычисление коэффициентов аппроксимирующего полинома m -й степени для функции $f(x)$, заданной таблично в $n+1$ точках, по методу наименьших квадратов требуются $\frac{n(m^2+5m+2)}{2} + \frac{1}{6}m \times$
 $\times (m^2+9m+8)$ операций сложения, $\frac{(n+1)(m^3+3m^2)}{2} +$
 $+\frac{1}{6}(m+1)(m^2+11m+12)$ операций умножения, $m+1$ операций деления и $m+1$ операций извлечения квадратных корней.

15. Взвешенное среднеквадратическое приближение. При составлении суммы σ по формуле (VI.47) предполагалось, что все точки x_i играют одинаковую роль: рассуждения проводились таким образом, как если бы выполнение всех равенств

$$P_m(x_i) - y_i = 0, \quad i = 0, 1, \dots, n,$$

было для нас одинаково важным. Однако может случиться, что некоторым из этих равенств по тем или иным причинам придается большее значение, другим — меньшее. Например, если данные y_i получены в результате измерений с помощью приборов, имеющих различную точность, то, конечно, следует отдавать предпочтение тем данным, которые получены с помощью более точных и надежных приборов. Математически это выражается в том, что сумма (VI.48) заменяется суммой более общего вида

$$\sigma = \sum_{i=0}^n \rho_i [P_m(x_i) - y_i]^2,$$

где каждое ρ_i — специальным образом подобранное положительное число, называемое весом в точке x_i (в некоторых случаях вес ρ_i нормируется так, чтобы $\sum_{i=0}^n \rho_i = 1$). Во всем остальном поступаем так же, как и в п. 1 настоящего параграфа. Заметим только, что система урав-

нений, из которой определяются коэффициенты a_k искомого полинома $P_m(x)$, в этом случае имеет вид

[illegible]

где

$$c_p = \sum_{i=0}^n \rho_i x_i^p \quad (p = 0, 1, \dots, 2m),$$

$$d_p = \sum_{i=0}^n \rho_i x_i^p y_i, \quad p = 0, 1, \dots, m.$$

16. Общая задача приближения по методу наименьших квадратов.

При использовании метода наименьших квадратов аппроксимирующая функция $P(x)$ не обязательно должна быть полиномом. Надо только, чтобы она была линейной комбинацией заданной системы линейно независимых функций ¹⁹.

$$P(x) = a_0 \varphi_0(x) + a_1 \varphi_1(x) + \dots + a_m \varphi_m(x). \quad (\text{VI.52})$$

Выражение (VI.52) называется обобщенным полиномом по рассматриваемой системе функций.

В п. 13, 14 параграфа VI.1 в качестве системы линейно независимых функций выступает система $1, x, x^2, \dots, x^n$. Системы линейно независимых функций

$$1, \cos x, \cos 2x, \dots, \cos mx, \\ \sin x, \sin 2x, \dots, \sin mx$$

обычно применяются при аппроксимации периодических функций.

Предполагаем, что функция $y = f(x)$ задана таблицей значений в отдельных точках $x_0, x_1, \dots, x_m$ отрезка $[a, b]$. Рассматриваем следующую задачу: среди обобщенных полиномов вида (VI.52) с $m \leq n$ найти такой, который минимизирует сумму

$$\sigma = \sum_{i=0}^n [P(x_i) - f(x_i)]^2. \quad (\text{VI.53})$$

Используя необходимые условия экстремума функции $\sigma = \sigma(a_0, a_1, \dots, a_m)$, получаем систему линейных уравнений для нахождения коэффициентов $a_0, a_1, \dots, a_m$. В этом случае система прини-

¹⁹ Напомним, что система функций $\varphi_0(x), \varphi_1(x), \dots, \varphi_m(x)$ называется линейно независимой на отрезке $[a, b]$, если тождество

$$c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_m\varphi_m(x) \equiv 0,$$

... , c_m — некоторые постоянные, когда $c_0 = c_1 = \dots = c_m = 0$, где $c_0, c_1, \dots$

имеет следующий вид:

$$\frac{1}{2} \frac{\partial \sigma}{\partial a_k} = \sum_{i=0}^n [a_0 \varphi_0(x_i) + a_1 \varphi_1(x_i) + \dots + a_m \varphi_m(x_i) - f(x_i)] \varphi_k(x_i) = 0, \quad k = 0, 1, \dots, m. \quad (\text{VI.54})$$

Вводя сокращенные обозначения $(\varphi, \psi) = \sum_{i=0}^n \varphi(x_i) \psi(x_i)$, систему (VI.54)

записываем в виде

[illegible]

Эта система имеет положительно определенную матрицу. Единственное решение $a_0 = a^*$, $a_1 = a_1^*$, ..., $a_m = a_m^*$ системы уравнений (VI.55) дает обобщенный полином $P^*(x) = a_0^* \varphi_0(x) + a_1^* \varphi_1(x) + \dots + a_m^* \varphi_m(x)$, обладающий минимальным квадратичным отклонением.

Система (VI.55) значительно упрощается, если φ_i , $i = 0, 1, \dots, m$, являются ортогональными полиномами, т. е.

$$(\varphi_j, \varphi_k) = \sum_{i=0}^n \varphi_j(x_i) \varphi_k(x_i) = 0, \quad j \neq k,$$

$$(\varphi_k, \varphi_k) = \sum_{i=1}^n \varphi_k^2(x_i) \neq 0.$$

17. Среднеквадратическое приближение функций нескольких пе-

ременных. До сих пор рассматривался метод наименьших квадратов только для функций от одной переменной. Принципиально ничего не меняется при переходе к функциям от нескольких переменных, однако резко возрастают объем вычислительной работы и сложность программ для ЭВМ.

Пусть функция $y = f(x)$, где x — точка $(p+1)$ -мерного векторного пространства, $x = (x^{(0)}, x^{(1)}, \dots, x^{(p)})$, задана таблично: для значений аргументов $x_0, x_1, \dots, x_n$ имеем соответствующий набор значений функции $y_0, y_1, \dots, y_n$. Будем рассматривать обобщенные полиномы вида

$$P(\mathbf{x}) = a_0 \varphi_0(\mathbf{x}) + a_1 \varphi_1(\mathbf{x}) + \dots + a_m \varphi_m(\mathbf{x}), \quad (\text{VI.56})$$

где $\varphi_0(x), \varphi_1(x), \dots, \varphi_m(x)$ — заданная система линейно независимых непрерывных функций (определение линейной независимости функций от одной переменной). Среди обобщенных полиномов вида (VI.56) будем находить такой, который доставляет минимум выражению

$$\sigma = \sum_{i=0}^n [P(\mathbf{x}_i) - y_i]^2.$$

Применяя необходимые условия экстремума функции $\sigma = \sigma(a_0, \dots, a_m)$, получаем систему линейных алгебраических уравнений вида (VI.55), в которой в этом общем случае

$$(\varphi_k, \varphi_l) = \sum_{i=0}^n \varphi_k(x_i) \varphi_l(x_i), \quad k, l = 0, 1, \dots, m,$$

$$(f, \varphi_k) = \sum_{i=0}^n y_i \varphi_k(x_i), \quad k = 0, 1, \dots, m.$$

Систему линейных алгебраических уравнений целесообразно решать методом квадратных корней.

Метод наименьших квадратов с системой линейно независимых функций

$$\varphi_0(x) = x^{(0)}, \varphi_1(x) = x^{(1)}, \dots, \varphi_n(x) = x^{(m)}, \quad (VI.57)$$

$$x = (x^{(0)}, x^{(1)}, \dots, x^{(m)})$$

часто применяется в математической статистике.

Коэффициенты системы (VI.55) в этом случае принимают следующий вид:

$$(\varphi_k, \varphi_l) = \sum_{i=0}^n x_i^{(k)} x_i^{(l)}, \quad k, l = 0, 1, \dots, m,$$

$$(f, \varphi_k) = \sum_{i=0}^n y_i x_i^{(k)}, \quad k = 0, 1, \dots, m.$$

Для построения аппроксимирующего полинома в конкретном случае (VI.57) необходимо произвести $\frac{n(m^3 + 3m + 4)}{2} + \frac{1}{6} m(m^2 + 9m + 8)$ операций сложения, $\frac{(n+1)(m^2 + 3m + 4)}{2} + \frac{1}{6} (m+1)(m^2 + 11m + 12)$ операций умножения, $m+1$ операций деления и $m+1$ операций извлечения квадратных корней.

VI.2. ЧИСЛЕННОЕ ДИФФЕРЕНЦИРОВАНИЕ И ИНТЕГРИРОВАНИЕ

1. Вопросы, возникающие при численном дифференцировании и интегрировании. В ряде случаев при решении задач на ЭВМ приходится проводить дифференцирование таблично заданных функций или вычислять необходимые интегралы. Так, в задаче (VI.1) (VI.4) помимо распределения температуры необходимо найти тепловое сопротивление, вычисляемое по формуле

$$R(x_1) = \frac{u(x_1, l_2 + 0) - u(x_1, l_2 - 0)}{Q} - p, \quad l_1 \leq x_1 \leq l_2, \quad (VI.58)$$

где p — заданное число и

$$Q = \int_0^{l_1} K_2(x_1, l_2) \frac{\partial u}{\partial x_2}(x_1, l_2) dx + \int_0^{l_1} K_1(x_1, 0) \frac{\partial u}{\partial x_1}(x_1, 0) dx. \quad (VI.59)$$

При решении задачи (VI.1) — (VI.4) методом сеток на ЭВМ получим приближение к решению y_k , поэтому для нахождения значения Q необходимо произвести численное дифференцирование функции y и в дальнейшем выполнить численное интегрирование.

Простейшие формулы численного дифференцирования получаем в результате дифференцирования интерполяционных формул

$$F(x) = P_n(x) + R_n(x),$$

где $P_n(x)$ — интерполяционная формула, $R_n(x)$ — остаточный член. Дифференцируя ее, получаем

$$F'(x) = P'_n(x) + R'_n(x).$$

Будем приближенно полагать, что

$$F'(x) = P'_n(x).$$

При таком предположении допускаем погрешность, равную $R'_n(x)$. Аналогично поступают при нахождении производных высших порядков. Используя, например, линейную интерполяцию (VI.7), получаем

$$\frac{dF}{dx} \sim \frac{dP_1}{dx} = \frac{f_{k+1} - f_k}{x_{k+1} - x_k}, \quad x_k \leq x \leq x_{k+1}. \quad (VI.60)$$

Оцениваем точность, с которой по формуле определяется значение производной от функции $F(x)$. Погрешность интерполяции в силу (VI.11) при $n = 1$ есть

$$R_1(x) = \frac{1}{2} F''(\xi(x)) (x - x_k)(x - x_{k+1}), \quad x_k \leq x \leq x_{k+1}. \quad (VI.61)$$

Дифференцируя это выражение, имеем

$$\frac{dR_1}{dx} = \frac{1}{2} F'''(\xi(x)) \xi'(x) (x - x_k)(x - x_{k+1}) +$$

$$+ \frac{1}{2} F''(\xi(x)) (2x - x_k - x_{k+1}).$$

Второе слагаемое в правой части порядка $x_{k+1} - x_k$, т. е. порядка h . Такова точность вычисления производной по формуле (VI.60). Исключением является центральная точка интервала $x = \frac{x_k + x_{k+1}}{2}$. В ней второе слагаемое обращается в нуль и, следовательно, по формуле (VI.60) получается значение производной в этой точке с точностью до h^2 . Попытка определения с помощью линейной интерполяции второй производной приводит к формуле

$$\frac{\partial^2 F}{\partial x^2} \sim \frac{d^2 P_1}{dx^2} = 0,$$

что, очевидно, непригодно. Это согласуется с тем, что

$$\frac{d^2 R_1}{dx^2} = F''(\xi(x)) + \dots,$$

т. е. в этом случае погрешность конечна, не убывает с уменьшением h . Для вычисления старших производных необходимо использовать интерполяцию более высокой степени.

Определим численно значение производной функции $y = \sin \pi x$ в точке $x = 1/3 \approx 0,333\ 333\ 333$, используя следующие данные:

x	0	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{2}$
y	0	$\frac{1}{2}$	$\frac{1\sqrt{2}}{2}$	1.

После применения интерполяционной формулы Ньютона, получен следующий результат (вычисления проводились на ЭВМ МИР-2):

$$y'(1/3) \approx 1,604\ 569\ 5.$$

Точное значение производной функции $y = \sin \pi x$ в точке $x = 1/3$ равняется $\pi/2 \approx 1,570\ 796\ 3$. Видно, что имеются сильные расхождения. Но это естественно, так как функции могут быть и очень близкими друг к другу, но иметь сильно различающиеся производные. Иными словами, это означает, что задача численного дифференцирования с применением интерполяционных формул является некорректной. Поэтому для ее решения естественно применить один из методов регуляризации, например метод регуляризованных сплайнов. Этот метод позволяет проводить численное дифференцирование табличных функций, значения которых заданы приближенно. Метод регуляризованных сплайнов рассмотрен в работе [24].

Теперь проанализируем вопросы, возникающие при численном интегрировании. Пусть требуется вычислить интеграл от функции, заданной значениями f_k в точках x_k , $k = 0, 1, \dots, N$. В простейшем случае можно использовать на каждом интервале (x_k, x_{k+1}) формулу линейного интерполирования (VI.7) и получить

$$\int_{x_k}^{x_{k+1}} P_1(x) dx = \frac{f_k + f_{k+1}}{2} (x_{k+1} - x_k).$$

Суммируя это выражение по всем интервалам, получаем способ вычисления интеграла. Для случая, когда расстояние между узлами постоянно ($x_{k+1} - x_k = h$), получаем квадратурную формулу трапеций.

$$\int_{x_0}^{x_N} P_1(x) dx = \left(\frac{1}{2} f_0 + f_1 + f_2 + \dots + f_{N-1} + \frac{1}{2} f_N \right) h. \quad (\text{VI.62})$$

Можно оценить точность, с которой по формуле (VI.62) вычисляется интеграл от функции $F(x)$. Погрешность интерполяции при $n = 1$ есть (VI.61). Отсюда

$$\left| \int_{x_k}^{x_{k+1}} R_1(x) dx \right| \leq \text{const} \max_{x_k \leq x \leq x_{k+1}} |F''| (x_{k+1} - x_k)^3.$$

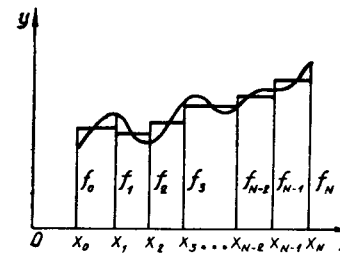


Рис. 19

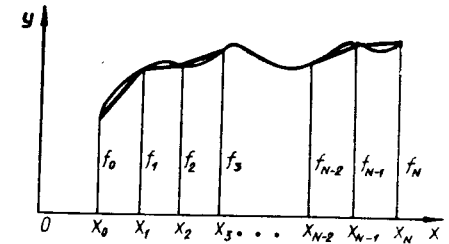


Рис. 20

Складывая эти неравенства по всем интервалам и учитывая, что $Nh = x_N - x_0$, имеем

$$\left| \int_{x_0}^{x_N} R_1(x) dx \right| \leq \text{const} \max_{x_0 \leq x \leq x_N} |F''| h^2,$$

т. е. формула трапеций (VI.62) имеет точность порядка h^2 .

Используя другие интерполяционные формулы, ниже получим и другие формулы для численного интегрирования. Так, например, используя $P_0(x) = f_k$, $x_k \leq x \leq x_{k+1}$, $k = 0, 1, 2, \dots$, получаем на каждом интервале x_k, x_{k+1} формулу прямоугольника

$$\int_{x_k}^{x_{k+1}} P_0(x) dx = f_k (x_{k+1} - x_k),$$

погрешность которой, как нетрудно убедиться, есть $\frac{1}{2} M_1 h^2$, где

$$(x_{k+1} - x_k) = h, \quad M_1 = \max_{x \in [x_k, x_{k+1}]} |F'|.$$

Суммируя последний интеграл по всем интервалам, получаем способ вычисления интеграла. Для случая, когда $x_{k+1} - x_k = h$, имеем

$$\int_{x_0}^{x_N} P_0(x) dx = h (f_0 + f_1 + \dots + f_{N-1}),$$

геометрическая интерпретация которого представлена на рис. 19. Геометрическая интерпретация формулы (VI.62) представлена на рис. 20. Для вычисления интеграла по формуле трапеций (VI.62) требуется произвести n операций сложения и три операции умножения.

2. Численное дифференцирование с применением формулы Ньютона. Пусть имеем функцию $y = f(x)$, заданную ее значениями в точках $x_0, x_1, \dots, x_n$ отрезка $[a, b]$. Для нее интерполяционная формула Ньютона с использованием разделенных разностей имеет вид

$$f(x) = P_n(x) + R_n(x), \quad (\text{VI.63})$$

где

$$P_n(x) = f(x_0) + f(x_0; x_1)(x - x_0) + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots + f(x_0; x_1; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1})$$

есть полином степени n и

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} q(x), \quad q(x) = \prod_{i=0}^n (x - x_i).$$

Значение $f^{(n+1)}(\xi)$, входящее в эту формулу, зависит от переменной x :

$$\xi \in (\alpha, \beta), \quad \alpha = \min \{x, x_0, x_1, \dots, x_n\}, \quad \beta = \max \{x, x_0, x_1, \dots, x_n\}.$$

Дифференцируя равенство (63), получаем

$$f'(x) = P'_n(x) + R'_n(x), \quad (\text{VI.64})$$

где

$$P'_n(x) = f(x_0; x_1) + [(x - x_0)'(x - x_1) + (x - x_0)] f(x_0; x_1; x_2) + \\ + [((x - x_0)(x - x_1))'(x - x_2) + (x - x_0)(x - x_1)] f(x_0; x_1; x_2; x_3) + \dots \\ \dots + [((x - x_0) \dots (x - x_{n-2}))'(x - x_{n-1}) + (x - x_0) \dots \\ \dots (x - x_{n-2})] f(x_0; x_1, \dots, x_n),$$

$$R'_n(x) = \frac{1}{(n+1)!} \left[f^{(n+1)}(\xi) \frac{dq(x)}{dx} + q(x) \frac{d}{dx} (f^{(n+1)}(\xi)) \right].$$

Для вычисления этим методом производной функции $y = f(x)$ в одной точке требуется произвести $n^2 + 5n - 4$ операций сложения, $3n - 3$ операций умножения и $\frac{n(n+1)}{2}$ операций деления.

Формула (VI.64) применяется в случае произвольных значений независимого переменного. Она значительно упрощается, если x принимает одно из значений $x_0, x_1, \dots, x_n$. Например, при $x = x_0$ имеем

$$f'(x_0) \simeq f(x_0; x_1) + (x_0 - x_1) f(x_0; x_1; x_2) + (x_0 - x_1)(x_0 - x_2) \times \\ \times f(x_0; x_1; x_2; x_3) + \dots + (x_0 - x_1)(x_0 - x_2) \dots \\ \dots (x_0 - x_{n-1}) f(x_0; x_1; \dots; x_n).$$

При этом допускаем ошибку, равную

$$R'_n(x_0) = \frac{\prod_{i=1}^n (x_0 - x_i)}{(n+1)!} f^{(n+1)}(\xi).$$

3. Применение формул Грегори — Ньютона. Пусть имеем функцию $y = f(x)$, заданную в равноотстоящих точках $x_0, x_1 = x_0 + h, x_2 = x_0 + 2h, \dots, x_n = x_0 + nh$ отрезка $[a, b]$ с помощью значений $y_i = f_i(x)$, $i = 0, 1, \dots, n$. Для вычисления на $[a, b]$ производных $y' = f'(x)$, $y'' = f''(x)$ и т. д. в начале таблицы применяется формула Грегори — Ньютона интерполирования вперед

$$f(x) = P_n(x) + R_n(x),$$

где

$$P_n(x) = y_0 + t \Delta y_0 + \frac{t(t-1)}{2!} \Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!} \Delta^3 y_0 + \dots \\ \dots + \frac{t(t-1) \dots (t-n+1)}{n!} \Delta^n y_0,$$

$$R_n(x) = \frac{t(t-1) \dots (t-n)}{(n+1)!} h^{n+1} f^{(n+1)}(\xi), \quad \xi \in (\alpha, \beta),$$

$$\alpha = \min \{x, x_0, \dots, x_n\}, \quad \beta = \max \{x, x_0, \dots, x_n\}, \quad t = \frac{x - x_0}{h}.$$

Так как

$$\frac{dy}{dx} = \frac{dy}{dt} \frac{dt}{dx},$$

то

$$P'_n(x) = \frac{1}{h} \left[\Delta y_0 + \frac{2t-1}{2} \Delta^2 y_0 + \frac{3t^2-6t+2}{6} \Delta^3 y_0 + \dots + \right. \\ \left. + \frac{[t(t-1) \dots (t-n+2)]'(t-n+1) + t(t-1) \dots (t-n+2)}{n!} \Delta^n y_0 \right].$$

Предполагая

$$f'(x) \simeq P'_n(x),$$

допускаем ошибку, равную

$$R'_n(x) = \frac{h^n}{(n+1)!} \left\{ f^{(n+1)}(\xi) \frac{d}{dt} [t(t-1) \dots (t-n)] + \right. \\ \left. + t(t-1) \dots (t-n) \frac{d}{dt} [f^{(n+1)}(\xi)] \right\}.$$

Для вычисления производной $y' = f'(x)$ в конце отрезка удобно применять формулу Грегори — Ньютона интерполирования назад. В этом случае аналогичным путем находим

$$f'(x) = P'_n(x) + R'_n(x),$$

где

$$P'_n(x) = \frac{1}{h} \left(\Delta y_{n-1} + \frac{2t+1}{2!} \Delta^2 y_{n-2} + \frac{3t^2+6t+2}{3!} \Delta^3 y_{n-3} + \dots \right. \\ \left. \dots + \frac{[t(t+1) \dots (t+n-2)]'(t+n-1) + t(t+1) \dots (t+n-2)}{n!} \Delta^n y_0 \right) = \\ = \frac{1}{h} \left(\nabla y_n + \frac{2t+1}{2!} \Delta^2 y_n + \frac{3t^2+6t+2}{3!} \nabla^3 y_n + \dots \right. \\ \left. \dots + \frac{[t(t+1) \dots (t+n-2)]'(t+n-1) + t(t+1) \dots (t+n-2)}{n!} \nabla^n y_n \right),$$

$$R'_n(x) = \frac{h^n}{(n+1)!} \left\{ f^{(n+1)}(\xi) \frac{d}{dt} [t(t+1) \dots (t+n)] + \right. \\ \left. + t(t+1) \dots (t+n) \frac{d}{dt} [f^{(n+1)}(\xi)] \right\}, \quad t = \frac{x - x_n}{h}.$$

При вычислении производной в одной точке с применением интерполяционной формулы Грегори — Ньютона требуется произвести $3n^2 + 9n$ операций сложения, $n^2 + 13n - 7$ операций умножения и $n^2 + n + 2$ операций деления.

4. Применение формулы Стирлинга. Пусть $x_{-n}, x_{-n+1}, \dots, x_{-1}, x_0, x_1, \dots, x_{n-1}, x_n$ — система равноотстоящих точек с шагом $x_{i+1} - x_i = h$, а $y_i = f(x_i)$ — соответствующие значения данной функции $y = f(x)$. Для численного дифференцирования вблизи середины отрезка $[x_{-n}, x_n]$ применяется формула, получаемая дифференцированием интерполяционной формулы Стирлинга:

$$f'(x) = P'_{2n}(x) + R'_{2n}(x), \quad (\text{VI.65})$$

где

$$\begin{aligned} P'_{2n}(x) = & \frac{1}{h} \left\{ \mu \delta y_0 + t \delta^2 y_0 + \frac{(t^2 - 1) + 2t^2}{3!} \mu \delta^3 y_0 + \right. \\ & + \frac{2t(t^2 - 1) + 2t^3}{4!} \delta^4 y_0 + \dots \\ & + \frac{\{t(t-1) \dots [t^2 - (n-2)^2]\}' [t^2 - (n-1)^2] + 2t^2(t^2 - 1) \dots [t^2 - (n-2)^2]}{(2n-1)!} \times \\ & \times \mu \delta^{2n-1} y_0 + \\ & + \frac{\{t^2(t^2 - 1) \dots [t^2 - (n-2)^2]\}' [t^2 - (n-1)^2] + 2t^3(t^2 - 1) \dots [t^2 - (n-2)^2]}{(2n)!} \times \\ & \times \delta^{2n} y_0 \left. \right\}, \end{aligned}$$

$$\begin{aligned} R'_{2n}(x) = & \frac{t^{2n+1}}{(2n+1)!} \left\{ f^{(2n+1)}(\xi) \frac{d}{dt} [t(t^2 - 1) \dots (t^2 - n^2)] + \right. \\ & + t(t^2 - 1) \dots (t^2 - n^2) \frac{d}{dt} [f^{(2n+1)}(\xi)] \left. \right\}, \quad \xi \in (x_{-n}, x_n). \end{aligned}$$

В частности, при $x = x_0$

$$\begin{aligned} f'(x_0) \simeq P'_{2n}(x_0) = & \frac{1}{h} \left\{ \mu \delta y_0 - \frac{1}{3!} \mu \delta^3 y_0 + \dots \right. \\ & \left. + \frac{(-1)^{n-1} ((n-1)!)^2}{(2n-1)!} \mu \delta^{2n-1} y_0 \right\}. \end{aligned}$$

При этом допускается ошибка

$$R'_{2n}(x_0) = \frac{(-1)^n h^{2n+1} (n!)^2}{(2n+1)!} f^{(2n+1)}(\xi).$$

Напомним еще, что центральные разности, входящие в формулу (VI.65), можно выражать через нисходящие разности:

$$\mu \delta y_0 = \frac{\Delta y_0 + \Delta y_{-1}}{2}, \quad \delta^2 y_0 = \Delta y_{-1}, \quad \mu \delta^3 y_0 = \frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2},$$

$$\delta^4 y_0 = \Delta^4 y_{-2}, \dots, \mu \delta^{2n-1} = \frac{\Delta^{2n-1} y_{-n} + \Delta^{2n-1} y_{-n+1}}{2}, \quad \delta^{2n} y_0 = \Delta^{2n} y_{-n}.$$

При вычислении производной в одной точке с применением интерполяционной формулы Стирлинга требуется произвести $3n^2 + 9n$ операций сложения, $n^2 + 13n - 7$ операций умножения и $n^2 + n + 2$ операций деления.

5. Интерполяционные квадратурные формулы. Пусть требуется проинтегрировать функцию $f(x)$, заданную таблично в точках $x_1, x_2, \dots, x_n$. Как следует из п. 1 параграфа VI.2, для интегрирования таблично заданной функции целесообразно сначала построить для подынтегральной функции интерполирующий полином, а затем его проинтегрировать.

В качестве интерполяционного полинома может быть использован, например полином Лагранжа (см. п. 2 параграфа VI.1). Тогда для подынтегральной функции $f(x)$ имеем

$$f(x) = P_{n-1}(x) + R_{n-1}(x), \quad (\text{VI.66})$$

где

$$P_{n-1}(x) = \sum_{i=1}^n \frac{\omega(x)}{(x-x_i) \omega'(x_i)} f(x_i), \quad \omega(x) = (x-x_1) \dots (x-x_n),$$

и $R_{n-1}(x)$ — остаточный член. Точное значение интеграла $\int_a^b f(x) dx$ будет

$$\int_a^b f(x) dx = \int_a^b P_{n-1}(x) dx + \int_a^b R_{n-1}(x) dx. \quad (\text{VI.67})$$

Если полином $P_{n-1}(x)$ достаточно хорошо приближает функцию $f(x)$ на всем отрезке $[a, b]$, то вторым членом в равенстве (VI.67) можно пренебречь и, следовательно, имеем

$$\int_a^b f(x) dx \simeq \sum_{i=1}^n A_i f(x_i), \quad (\text{VI.68})$$

где

$$A_i = \int_a^b \frac{\omega(x)}{(x-x_i) \omega'(x_i)} dx. \quad (\text{VI.69})$$

Квадратурные формулы вида (VI.68) с коэффициентами (VI.69) называются интерполяционными. Для того чтобы квадратурная формула была интерполяционной, необходимым и достаточным условием для этого является следующее: она должна быть точной для всевозможных полиномов степени, не превышающей $n-1$ [25]. Если пределы интегрирования a и b являются узлами интерполирования, то квадратурная интерполяционная формула называется замкнутого типа, в противном случае — открытого типа.

6. Квадратурные формулы Ньютона — Котеса. Пусть для функции $y = f(x)$ требуется вычислить интеграл $\int_a^b f(x) dx$. Выбрав шаг $h = \frac{b-a}{n}$, разобьем отрезок $[a, b]$ с помощью равноотстоящих точек $x_0 = a, x_1 = x_0 + h, x_2 = x_0 + 2h, \dots, x_{n-1} = x_n + (n-1)h, x_n = b$

на n равных частей. И пусть $y_i = f(x_i)$, $i = 0, 1, \dots, n$. Для нахождения квадратурной формулы используем интерполяционный полином Лагранжа, который в данном случае принимает следующий вид:

$$h_n(x) = \sum_{i=0}^n \frac{(-1)^{n-1}}{i!(n-i)!} \frac{t(t-1)\dots(t-n)}{t-i} y_i, \quad t = \frac{x-x_0}{h}.$$

Заменяя подынтегральную функцию $y = f(x)$ этим полиномом и учитывая, что $dx = hdt$, получаем следующую квадратурную формулу:

$$\int_a^b f(x) dx \approx (b-a) \sum_{k=0}^n H_k y_k, \quad (\text{VI.70})$$

где $h = \frac{b-a}{n}$, $y_k = f(a + kh)$, $k = 0, 1, \dots, n$,

$$H_k = \frac{1}{n} \frac{(-1)^{n-1}}{k!(n-k)!} \int_0^n \frac{t(t-1)\dots(t-n)}{t-k} dt, \quad k = 0, 1, \dots, n. \quad (\text{VI.71})$$

Постоянные H_k называют коэффициентами Котеса, для которых справедливы соотношения

$$\sum_{k=0}^n H_k = 1, \quad H_k = H_{n-k}.$$

Ошибка формулы (VI.70)

$$R = -\mu_n (b-a)^{2n+1} f^{(2n)}(\xi), \quad \xi \in (a, b), \quad \mu_n = \frac{(n!)^4}{[(2n)!]^2(2n+1)}.$$

Пример 9. Вычислить интеграл $\int_0^1 \frac{dx}{1+x}$, применяя формулу Ньютона — Котеса с семью ординатами ($n = 6$). В этом случае коэффициенты Котеса следующие:

$$H_0 = H_6 = \frac{41}{840} \approx 0,048\,809\,524, \quad H_1 = H_5 = \frac{9}{35} \approx 0,257\,142\,86,$$

$$H_2 = H_4 = \frac{9}{280} \approx 0,032\,142\,857, \quad H_3 = \frac{34}{105} \approx 0,323\,809\,52.$$

Решение. Полагая $h = \frac{1-0}{6} = \frac{1}{6}$ и вычисляем значения функции $y = \frac{1}{1+x}$ в точках $x = 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1$. В результате имеем

$$y_0 = y(0) = 1, \quad y_1 = y\left(\frac{1}{6}\right) = \frac{6}{7}, \quad y_2 = y\left(\frac{1}{3}\right) = \frac{3}{4}, \quad y_3 = y\left(\frac{1}{2}\right) = \frac{2}{3},$$

$$y_4 = y\left(\frac{2}{3}\right) = \frac{3}{5}, \quad y_5 = y\left(\frac{5}{6}\right) = \frac{6}{11}, \quad y_6 = y(1) = \frac{1}{2}.$$

Применяя формулу Ньютона — Котеса, находим

$$\int_0^1 \frac{dx}{1+x} \approx H_0 y_0 + H_1 y_1 + H_2 y_2 + H_3 y_3 + H_4 y_4 + H_5 y_5 + H_6 y_6 \approx 0,693\,148\,07.$$

Точное значение интеграла следующее: $\int_0^1 \frac{dx}{1+x} = \ln 2 = 0,693\,15 \dots$

Для вычисления интеграла по формуле Ньютона — Котеса требуется произвести $n+1+(n+1)k+1$ операций сложения, $n+(n+1)l+2$ операций умножения и $(n+1)q$ операций деления, где k, l и q — соответственно числа операций сложения, умножения и деления, которые затрачиваются при вычислении подынтегральной функции $y = f(x)$ в одном узле.

Формула Ньютона — Котеса точна для полиномов степени $n+1$ при n четном и степени n при n нечетном [25].

При $n = 1$ имеем $H_0 = H_1 = \frac{1}{2}$ и

$$\int_a^b f(x) dx \approx \frac{b-a}{2} (y_0 + y_1).$$

Это известная формула трапеций. Для нее остаточный член

$$R = -\frac{(b-a)^3}{12} f''(\xi), \quad \xi \in (a, b).$$

Если $n = 2$, то коэффициенты Котеса принимают значения $H_0 = H_2 = \frac{1}{6}$, $H_1 = \frac{2}{3}$ и квадратурная формула (VI.70) дает формулу Симпсона

$$\int_a^b f(x) dx \approx \frac{b-a}{2} (y_0 + 4y_1 + y_2).$$

Ошибка формулы Симпсона

$$R = -\frac{h^5}{90} f^{(4)}(\xi), \quad \xi \in (a, b).$$

При разбиении отрезка $[a, b]$ на $2m$ равных частей можно получить обобщенную формулу Симпсона

$$\int_a^b f(x) dx \approx \frac{b-a}{6m} (y_0 + 4y_1 + 2y_2 + \dots + 2y_{2m-2} + 4y_{2m-1} + y_{2m}).$$

Ее ошибка имеет вид

$$R = -\frac{h^5}{90} m f^{(4)}(\xi), \quad \xi \in (a, b).$$

Пример 10. Вычислить интеграл

$$\mathcal{J} = \int_0^1 \frac{dx}{1+x^2}$$

по обобщенной формуле Симпсона, разбив отрезок интегрирования на 10 равных частей.

Решение. В данном случае имеем

$$\begin{aligned} y_0 &= y(0) = 1; \quad y_1 = y(0,1) \approx 0,990\,099\,00; \quad y_2 = y(0,2) \approx 0,961\,588\,46; \\ y_3 &= y(0,3) \approx 0,917\,431\,19; \quad y_4 = y(0,4) \approx 0,862\,068\,96; \quad y_5 = y(0,5) \approx 0,8; \\ y_6 &= y(0,6) \approx 0,735\,294\,11; \quad y_7 = y(0,7) \approx 0,671\,140\,93; \quad y_8 = y(0,8) \approx 0,609\,756\,09; \\ y_9 &= y(0,9) \approx 0,552\,486\,18; \quad y_{10} = y(1) \approx 0,5. \end{aligned}$$

Применяя формулу Симпсона, получим

$$\begin{aligned} \mathcal{J} &= \int_0^1 \frac{dx}{1+x^2} \approx \frac{1}{30} (y_0 + y_1 + 2y_2 + 4y_3 + 2y_4 + 4y_5 + 2y_6 + 4y_7 + 2y_8 + \\ &\quad + 4y_9 + y_{10}) \approx 0,785\,398\,15. \end{aligned}$$

Точное значение интеграла следующее: $\mathcal{J} = \frac{\pi}{4} = 0,785\,398\,16\dots$

Для вычисления интеграла по формуле Симпсона требуется произвести $2m+1+(2m+1)k$ операций сложения, $2m+1+(2m+1)l$ операций умножения и $1+(2m+1)q$ операций деления, где k, l и q соответственно числа операций сложения, умножения и деления, которые затрачиваются при вычислении подынтегральной функции $y = f(x)$ в одном узле.

7. Формулы Чебышева для численного интегрирования. Будем сначала рассматривать определенный интеграл от функции $f(x)$ на отрезке $[-1, +1]$. По аналогии с формулой Ньютона — Котеса можно получить бесчисленное множество приближенных формул вида

$$\int_{-1}^1 f(t) dt = \sum_{i=1}^n B_i f(t_i), \quad (\text{VI.72})$$

где B_i — постоянные, t_i — точки отрезка $[-1, 1]$.

Чебышев предложил при построении квадратурных формул типа (VI.72) задавать не точки t_i , а постоянные B_i , а точки $t_1, t_2, \dots, t_n$ выбирать так, чтобы формула была точной для любого алгебраического полинома степени, не выше n . При этом коэффициенты B_i задаются так, чтобы формула была возможно проще для вычислений, что, очевидно, будет тогда, когда все коэффициенты между собою равны, т. е. $B_1 = B_2 = \dots = B_n = B$ и

$$\int_{-1}^1 f(t) dt \approx B \sum_{i=1}^n f(t_i).$$

Учитывая, что при $f(t) \equiv 1$ эта формула должна быть точной, находим

$$2 = Bn, \quad B = \frac{2}{n},$$

$$\int_{-1}^1 f(t) dt \approx \frac{2}{n} \sum_{i=1}^n f(t_i). \quad (\text{VI.73})$$

Для определения абсцисс t_i используем условие, что формула (VI.73) должна быть точной для полиномов $t, t^2, \dots, t^n$. Подставляя эти функции в формулу (VI.73), получаем систему уравнений

$$\begin{aligned} t_1 + t_2 + \dots + t_n &= 0, \\ t_1^2 + t_2^2 + \dots + t_n^2 &= \frac{n}{3}, \\ t_1^3 + t_2^3 + \dots + t_n^3 &= 0, \\ t_1^5 + t_2^5 + \dots + t_n^5 &= \frac{n}{5}, \\ &\dots \dots \dots \\ t_1^n + t_2^n + \dots + t_n^n &= \frac{n(1 - (-1)^{n+1})}{2(n+1)}, \end{aligned} \quad (\text{VI.74})$$

из которой могут быть определены неизвестные $t_i, i = 1, 2, \dots, n$. Известно, что система (VI.74) при $n = 8$ и $n \geq 10$ не имеет действительных решений.

Для произвольного отрезка интегрирования имеем

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b+a}{2} + \frac{b-a}{2}t\right) dt.$$

Применяя к последнему интегралу квадратурную формулу (VI.73), получаем

$$\int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=1}^n f(x_i),$$

где $x_i = \frac{b+a}{2} + \frac{b-a}{2}t_i$, и $t_i, i = 1, 2, \dots, n$, — корни системы (VI.74).

Пример 11. Вычислить интеграл

$$\mathcal{J} = \int_0^1 \frac{dx}{1+x}$$

по формуле Чебышева с семью ординатами ($n = 7$).

Решение. В этом случае $f(x) = \frac{1}{1+x}$ и

$$\mathcal{J} \approx \frac{1}{7} [f(x_1) + f(x_2) + f(x_3) + f(x_4) + f(x_5) + f(x_6) + f(x_7)],$$

где

$$x_1 = \frac{1}{2} + \frac{1}{2}t_1 \approx \frac{1}{2} + \frac{1}{2}(-0,883\,862) = 0,058\,069\,00,$$

$$x_2 = \frac{1}{2} + \frac{1}{2}t_2 \approx \frac{1}{2} + \frac{1}{2}(-0,529\,657) = 0,235\,171\,50,$$

$$x_3 = \frac{1}{2} + \frac{1}{2} t_3 \approx \frac{1}{2} + \frac{1}{2} (-0,323\,912) = 0,338\,044\,00,$$

$$x_4 = \frac{1}{2} + \frac{1}{2} t_4 \approx \frac{1}{2} + \frac{1}{2} \cdot 0 = 0,5,$$

$$x_5 = 1 - x_3 \approx 0,661\,956\,00,$$

$$x_6 = 1 - x_2 \approx 0,764\,828\,50,$$

$$x_7 = 1 - x_1 \approx 0,941\,931\,00.$$

Находим теперь соответствующие значения функции $f(x)$:

$$f(x_1) = \frac{1}{1+x_1} \approx 0,945\,117\,95,$$

$$f(x_2) = \frac{1}{1+x_2} \approx 0,809\,604\,17,$$

$$f(x_3) = \frac{1}{1+x_3} \approx 0,747\,359\,58,$$

$$f(x_4) = \frac{1}{1+x_4} \approx 0,666\,666\,67,$$

$$f(x_5) = \frac{1}{1+x_5} \approx 0,601\,700\,65,$$

$$f(x_6) = \frac{1}{1+x_6} \approx 0,566\,627\,30,$$

$$f(x_7) = \frac{1}{1+x_7} \approx 0,514\,951\,36.$$

$$\mathcal{I} \approx \frac{1}{7} (0,945\,117\,95 + 0,809\,604\,17 + 0,747\,359\,58 + 0,666\,666\,67 +$$

$$+ 0,601\,700\,65 + 0,566\,627\,30 + 0,514\,951\,36) \approx 0,693\,146\,81.$$

Точное значение интеграла следующее: $\mathcal{I} = \ln 2 = 0,693\,147\,180\,5\dots$

Для вычисления интеграла по формуле Чебышева необходимо произвести $(2+k)n+1$ операций сложения, $(1+l)n+3$ операций умножения и $1+nq$ операций деления, где k , l и q соответственно числа операций сложения, умножения и деления, которые затрачиваются при вычислении подынтегральной функции $y = f(x)$ в одном узле.

8. Метод квадратуры Гаусса. Гаусс предложил в квадратурной формуле

$$\int_a^b f(x) dx \approx A_1 f(x_1) + A_2 f(x_2) + \dots + A_n f(x_n) \quad (\text{VI.75})$$

не считать заданными как точки x_i , так и коэффициенты A_i , а определять их так, чтобы квадратурная формула была точной для всех полиномов $Q(x)$ наивысшей возможной степени N . Легко видеть, что $N = 2n - 1$.

Для того чтобы формула (VI.75) была точной для всех полиномов степени, не превышающей $2n - 1$, необходимо и достаточно [26], чтобы она была интерполяционной и полином $\omega_n(x) = (x - x_1)(x - x_2)\dots(x - x_n)$ был ортогонален ко всем полиномам $Q(x)$ степени, меньшей n ,

$$\int_a^b \omega_n(x) Q(x) dx = 0.$$

Полином $\omega_n(x)$ будет ортогонален ко всем полиномам степени, меньшей n , тогда и только тогда, если он будет ортогонален к полиномам $1, x, \dots, x^{n-1}$:

$$\int_a^b x^k \omega_n(x) dx = 0, \quad k = 0, 1, \dots, n-1. \quad (\text{VI.76})$$

Таким образом, для нахождения квадратурной формулы Гаусса: определяем полином $\omega_n(x)$ с действительными различными корнями $x_1, x_2, \dots, x_n \in [a, b]$, который удовлетворяет условиям (VI.76); для найденных $x_1, \dots, x_n$ определяем коэффициенты $A_1, \dots, A_n$, используя при этом условие того, что формула (VI.75) должна быть интерполяционной.

Пусть отрезком интегрирования является $[-1, 1]$. Известно [3, 25], что системой полиномов, обладающих на отрезке $[-1, 1]$ свойством (VI.76), являются полиномы Лежандра

$$P_n(t) = \frac{1}{2^n n!} \frac{d^n (t^2 - 1)^n}{dt^n}, \quad n = 0, 1, 2, \dots$$

Все корни полинома $P_n(t)$ вещественные, различные и расположены на отрезке $[-1, 1]$ симметрично относительно точки $t = 0$.

Используя корни $t_1, t_2, \dots, t_n$ полинома Лежандра $P_n(t)$, получаем следующую квадратурную формулу Гаусса:

$$\int_{-1}^1 f(t) dt \approx \sum_{i=1}^n A_i f(t_i), \quad (\text{VI.77})$$

где коэффициенты A_i определяются по формуле

$$A_i = \frac{2}{(1 - t_i^2) [P'_n(t_i)]^2}, \quad i = 1, 2, \dots, n. \quad (\text{VI.78})$$

Из формулы (VI.78) видно, что для корней t_i и $-t_i$ соответствующие коэффициенты равны.

Выведем квадратурную формулу Гаусса для случая трех ординат ($n = 3$). Полином Лежандра третьей степени $P_3(t) = \frac{1}{2}(5t^3 - 3t)$

имеет такие корни: $t_1 = -\sqrt{\frac{3}{5}}$, $t_2 = 0$, $t_3 = \sqrt{\frac{3}{5}}$. Применяя формулу (VI.78), находим соответствующие коэффициенты A_i , $i = 1, 2, 3$:

$$P'_3(t) = \frac{1}{2}(15t^2 - 3),$$

$$A_1 = \frac{2}{\left(1 - \frac{3}{5}\right) \left[\frac{1}{2} \left(15 \frac{3}{5} - 3\right)\right]^2} = \frac{5}{9} = A_3,$$

$$A_2 = \frac{2}{(1-0) \left[\frac{1}{2} (15 \cdot 0 - 3)\right]^2} = \frac{8}{9}.$$

Следовательно,

$$\int_{-1}^1 f(t) dt \simeq \frac{1}{9} \left[5f\left(-\sqrt{\frac{5}{3}}\right) + 8f(0) + 5f\left(\sqrt{\frac{5}{3}}\right) \right].$$

На любом конечном отрезке $[a, b]$ формула Гаусса будет иметь вид

$$\int_a^b f(x) dx \simeq \frac{b-a}{2} \sum_{i=1}^n A_i f(x_i), \quad (\text{VI.79})$$

где

$$x_i = \frac{b+a}{2} + \frac{b-a}{2} t_i, \quad i = 1, 2, \dots, n,$$

t_i — нули полинома Лежандра $P_n(t)$. Коэффициенты A_i вычисляются по формуле (VI.78).

Остаточный член формулы (VI.79) выражается следующим образом:

$$R = \frac{(b-a)^{2n+1} (n!)^4 f^{(2n)}(\xi)}{[(2n)!]^3 (2n+1)}.$$

Пример 12. Вычислить интеграл

$$\mathcal{J} = \int_1^2 \frac{dx}{x},$$

применяя формулу Гаусса с пятью ординатами ($n=5$) и используя следующие приближенные значения элементов формулы Гаусса: $t_1 \simeq -0,906\,179\,85$; $t_2 = -0,538\,469\,31$; $t_3 = 0$; $t_4 \simeq 0,538\,469\,31$; $t_5 \simeq 0,906\,179\,85$; $A_1 = A_5 = 0,236\,926\,88$; $A_2 = A_4 = 0,478\,628\,68$; $A_3 = 0,568\,888\,89$.

Решение. Находим значения абсцисс формулы Гаусса для $a=1, b=2$:

$$x_1 = \frac{3}{2} + \frac{1}{2} t_1 \approx \frac{3}{2} + \frac{1}{2} (-0,906\,179\,85) \approx 1,046\,910\,03,$$

$$x_2 = \frac{3}{2} + \frac{1}{2} t_2 \approx \frac{3}{2} + \frac{1}{2} (-0,538\,469\,31) \approx 1,230\,765\,34,$$

$$x_3 = \frac{3}{2} + \frac{1}{2} t_3 = \frac{3}{2} + \frac{1}{2} \cdot 0 = 1,5,$$

$$x_4 = 3 - x_2 \approx 1,769\,234\,66,$$

$$x_5 = 3 - x_1 \approx 1,953\,089\,92.$$

Вычисляем теперь соответствующие значения функции $y = f(x) = 1/x$

$$f(x_1) = \frac{1}{x_1} \approx 0,955\,191\,88, \quad f(x_2) = \frac{1}{x_2} \approx 0,812\,502\,57, \quad f(x_3) = \frac{1}{x_3} \approx$$

$$\approx 0,666\,666\,67, \quad f(x_4) = \frac{1}{x_4} \approx 0,565\,216\,15, \quad f(x_5) = \frac{1}{x_5} \approx 0,512\,009\,19.$$

Теперь по формуле Гаусса имеем

$$\mathcal{J} = \frac{b-a}{2} \sum_{i=1}^5 A_i f(x_i) \approx \frac{1}{2} [0,236\,926\,88 (0,955\,191\,88 + 0,512\,009\,19) + \\ + 0,478\,628\,68 (0,812\,502\,57 + 0,565\,216\,15) + 0,568\,888\,89 + 0,666\,666\,67] \approx \\ \approx 0,693\,147\,16.$$

Точное значение интеграла следующее: $\mathcal{J} = \ln 2 = 0,693\,147\,180\,5\dots$

Для вычисления интеграла по формуле Гаусса требуется произвести $2n+1+nk$ операций сложения, $2+3+nl$ операций умножения и nq операций деления, где k, l и q — соответственно числа операций сложения, умножения и деления, которые затрачиваются при вычислении подынтегральной функции в одном узле.

В ряде задач возникает необходимость приближенной замены функции $f(x)$, заданной на всем отрезке $[a, b]$, или во всяком случае в отдельных его точках $x_i, i=0, 1, \dots, n$, функцией $P(x)$ некоторого класса, значения которой в точках x_i совпадают с соответствующими значениями функции $f(x)$. Так возникают задачи интерполирования функции. В отдельных случаях требуется, чтобы заданные значения в узлах интерполирования приобретала не только интерполирующая функция, но и ее производные.

В принципе произвольная функция $f(x)$, совпадающая с $P(x)$ в узлах интерполирования, может как угодно отличаться от него в остальных точках. Но если $f(x)$ обладает на $[a, b]$ непрерывными производными до n -го порядка включительно и производная $f^{(n)}(x)$ дифференцируема на $[a, b]$, то

$$f(x) - P(x) = R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-x_0)(x-x_1)\dots(x-x_n),$$

где $x_0 < \xi < x_n$.

Остаточный член интерполирования $R_n(x)$ для заданной функции $f(x)$ зависит от полинома $(x-x_0)(x-x_1)\dots(x-x_n)$, который полностью определяется узлами интерполирования. В некоторых случаях имеется возможность выбирать узлы интерполирования на свое усмотрение и увеличивать точность интерполирования. Так, если в качестве узлов интерполирования взять нули полинома Чебышева $T_{n+1}(x) = \cos[n \arccos x]$, $|x| \leq 1$, то погрешность интерполирования на отрезке $[-1, 1]$ для данной функции $f(x)$ будет наименьшей. В этом случае остаточный член $R_n(x)$ может быть оценен следующим образом:

$$R_n(x) \leq \frac{M_{n+1}}{2^n (n+1)!},$$

где $M_{n+1} = \max_{x \in [a, b]} |f^{(n+1)}(x)|$. Если интерполирование производится

на произвольном отрезке $[a, b]$, то его можно перевести в отрезок $[-1, 1]$ линейной заменой переменного.

Для приближения таблично заданной функции с большим числом узлов иногда применяют метод минимакса. Основная идея этого метода состоит в том, чтобы уменьшить максимальное отклонение заданной функции в узлах от приближающей до минимума функции. Изложение теории и алгоритмов по этому методу имеется в работе [24].

В зависимости от свойств интерполируемой функции и положения заданных точек на отрезке интерполирования применяют различные интерполяционные формулы.

Если заданы равноотстоящие узлы интерполирования и значения аргумента расположены близко к левому или правому концу отрезка интерполирования, то в этом случае можно использовать формулы Грегори — Ньютона для интерполирования «вперед» (в окрестности левого конца) или назад (в окрестности правого конца).

Если задано нечетное число равноотстоящих узлов интерполирования и значения аргумента расположены близко к центру отрезка интерполирования, то целесообразно использовать формулу Стирлинга.

Если заданы неравноотстоящие узлы интерполирования и значение аргумента расположено в центре отрезка интерполирования, то целесообразно использовать формулу Лагранжа.

Если табулированы не только функция, но и ее производные вплоть до некоторого порядка, то для интерполирования такой функции и ее производных можно использовать интерполяционный полином Эрмита.

Широко применяют также интерполирование кусочно-аналитическими функциями (сплайнами). Наиболее важный представитель этого класса, по-видимому, — кубический сплайн. Он является кусочно-кубической функцией, которая обладает непрерывными первой и второй производными на всем отрезке интерполирования [81, 88].

Если интерполирующая функция $f(x)$ — периодическая, то можно интерполирующую функцию $P(x)$ искать в классе тригонометрических полиномов. Если интерполируемая функция обратится в бесконечность в заданных точках или вблизи них, то $P(x)$ целесообразно искать в классе рациональных функций [45].

В случае приближенного задания значений функции в узлах области иногда для приближения функций целесообразно использовать метод наименьших квадратов или среднеквадратичное приближение функций [2, 3, 99]. Среднеквадратичная аппроксимация функций — это нахождение для заданной функции такой другой функции из некоторого класса, для которой среднеквадратичное отклонение от данной функции минимально. Среднеквадратичным отклонением называют усреднение с некоторым весом по заданному множеству точек квадрата разности заданной и аппроксимирующей функций.

Помимо среднеквадратичной аппроксимации функций применяют равномерную, в которой условием минимизации является условие равномерной нормы отклонения [107, 108, 111, 113, 124]. В отличие от среднеквадратичной аппроксимации задача равномерной аппроксимации допускает точное прямое решение лишь в немногих частных случаях. В общей постановке она требует привлечения итерационных

численных методов. При переходе к аппроксимации функций нескольких переменных резко возрастает объем арифметических операций, необходимых для решения этой задачи. Таким образом, вместо таблично заданной функции, используя интерполяцию или аппроксимацию, можно получить аналитически заданную функцию.

Один из способов вычисления производных от функции $f(x)$, заданной таблицей ее значений в $n + 1$ узлах $x_i, i = 0, 1, \dots, n$, состоит в следующем: функцию $f(x)$ на интересующем нас отрезке заменяют интерполирующей функцией $P(x)$ и считают, что m -я производная при $x_0 \leq x \leq x_n$ определяется соотношением

$$f^{(m)}(x) = \lim_{h \rightarrow 0} \frac{f^{(m-1)}(x+h) - f^{(m-1)}(x)}{h} \approx P^{(m)}(x).$$

Выбор интерполяционной формулы $P(x)$ зависит от того, какая дана система узлов сетки для $f(x)$ и при каких значениях x нужно вычислять производные.

Задача отыскания производной $f'(x)$ по экспериментальной случайной функции $\hat{f}(x)$ значительно отличается от задачи дифференцирования функций, для которых известны точные данные. В этом случае наблюдения не свободны от случайных значительных ошибок, а отношения $[f(x+h) - f(x)]/h$ очень чувствительны даже к небольшим ошибкам, если h становится достаточно малым, поэтому обычные формулы численного дифференцирования могут сильно исказить результаты.

Задача восстановления производной по функции, заданной экспериментально, принадлежит к числу некорректно поставленных задач. Поэтому восстанавливать производную можно, используя метод регуляризации А. Н. Тихонова. Пусть $f(x)$ непрерывно дифференцируема на отрезке (x_0, x_n) , тогда ее производная по определению удовлетворяет интегральному уравнению Вольтерра первого рода

$$f(x) = \int_a^x f'(s) ds + f(d), \quad x_0 \leq x, \quad d \leq x_n,$$

для решения которого и применяют метод регуляризации.

Методам численного дифференцирования посвящены работы [2, 3, 22, 49, 70].

Большинство применяемых в настоящее время способов численного вычисления определенных интегралов основано на замене интегрируемой функции $f(x)$ простой и легко интегрируемой функцией. Эта замена, как правило, дает тем большую точность, чем выше порядок дифференцируемости функции $f(x)$ и чем более «гладко» ее изменение. Поэтому, прежде чем решать вопрос о выборе квадратурной формулы, необходимо составить представление о поведении функции и ее производных. Представление о поведении функции может, например, подсказать, что промежуток интегрирования следует разбить на части и к интегралу по каждой части промежутка применить свою квадратурную формулу. Численные методы вычисления интегралов и их теоретическое обоснование рассмотрены в работах [2, 25, 26, 43, 45, 48, 49, 50, 57, 71, 138, 158].

1. Абрамов А. А. Некоторые замечания о нахождении собственных значений и собственных векторов матриц, возникающих при применении метода Рунге или сеток // Журн. вычисл. математики и мат. физики.— 1967.— 7, № 3.— С. 644—647.
2. Алберг Дж., Нильсон Э., Уолли Дж. Теория сплайнов и ее приложения.— М.: Мир, 1972.— 320 с.
3. Ахизер Н. И. Лекции по теории аппроксимации.— М.: Наука, 1965.— 107 с.
4. Бате К., Вилсон Е. Численные методы анализа и метод конечных элементов.— М.: Стройиздат, 1982.— 447 с.
5. Бахвалов Н. С. Численные методы.— М.: Наука, 1973.— Т. 1.
- 6-7. Березин И. С., Жидков Н. П. Методы вычислений.— М.: Физматгиз, 1959. Т. 1—2.
8. Беллман Р. Динамическое программирование.— М.: Изд-во иностр. лит., 1960.— 400 с.
9. Бурдаков О. П. Некоторые глобально сходящиеся модификации метода Ньютона для решения систем нелинейных уравнений // Докл. АН СССР.— 1980.— 254, № 3.— С. 521—523.
10. Бурдаков О. П. Устойчивые варианты метода секущих для решения систем уравнений // Журн. вычисл. математики и мат. физики.— 1983.— 23, № 5.— С. 1027—1040.
11. Васильев Ф. П. Численные методы решения экстремальных задач.— М.: Наука, 1983.— 384 с.
12. Воеводин В. В. О применении метода спуска для нахождения всех корней алгебраического многочлена // Журн. вычисл. математики и мат. физики.— 1961.— 1, № 2.— С. 187—195.
13. Воеводин В. В. Ошибки округления и устойчивость в прямых методах линейной алгебры.— М.: Изд-во Моск. ун-та, 1969.— 153 с.
14. Воеводин В. В. Линейная алгебра.— М.: Наука, 1974.— 336 с.
15. Воеводин В. В. Вычислительные основы линейной алгебры.— М.: Наука, 1977.— 303 с.
16. Гавурин М. К. О плохо обусловленных системах линейных алгебраических уравнений // Журн. вычисл. математики и мат. физики.— 1962.— 2, № 3.— С. 387—397.
17. Глушков В. М., Молчанов И. Н., Николенко Л. Д. О наборе программ для решения систем линейных алгебраических уравнений на машинах серии МИР // Кибернетика.— 1968.— № 6.— С. 1—6.
18. Глушков В. М., Молчанов И. Н., Брусникин Б. Н. и др. Программное обеспечение ЭВМ МИР-1 и МИР-2.— Киев: Наук. думка, 1976.— Т. 1.
19. Гольдштейн Е. Г., Юдин Д. Б. Задачи линейного программирования транспортного типа.— М.: Наука, 1969.— 384 с.
20. Гончаров В. Л. Теория интерполирования и приближения функций.— М.: Госиздат, 1954.— 325 с.
21. Гордонова В. И., Морозов В. А. Численные алгоритмы выбора параметров в методе регуляризации // Журн. вычисл. математики и мат. физики.— 1973.— 13, № 3.— С. 539—545.
22. Гутер Р. С., Овчинский Б. В. Элементы численного анализа и математической обработки результатов опыта.— М.: Наука, 1970.— 432 с.
23. Даниц Дж. Линейное программирование, его применение и обобщения.— М.: Прогресс, 1966.— 597 с.
24. Демидович Б. П., Марон И. А. Основы вычислительной математики.— М.: Физматгиз, 1960.— 660 с.
25. Демидович Б. П., Марон И. А., Шувалова Э. З. Численные методы анализа.— М.: Физматгиз, 1960.— 367 с.
26. Демьянов В. Ф., Малоземов В. Н. Введение в минимакс.— М.: Наука, 1972.— 368 с.
27. Джеффрис Г., Свирле Б. Методы математической физики.— М.: Мир, 1970.— Вып. 2.— 352 с.
28. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений.— М.: Мир, 1984.— 333 с.
29. Дьяконов Е. Г. Разностные методы решения краевых задач.— М.: Изд-во Моск. ун-та, 1971—1972.— Вып. 1—2.— 218 с.
30. Дьяконов Е. Г. Модифицированные итерационные алгоритмы в задачах на собственные значения // Вычисл. методы линейн. алгебры.— 1978.— Вып. 3.— С. 39—61.
31. Евтушенко Ю. Г. Методы решения экстремальных задач и их применение в системах оптимизации.— М.: Наука, 1982.— 432 с.
32. Ермолов Ю. М. Методы стохастического программирования.— М.: Наука, 1976.— 239 с.
33. Зойтендейк Г. Методы возможных направлений.— М.: Изд-во иностр. лит., 1963.— 176 с.
34. Икрамов Х. Д. Разреженные матрицы // Итоги науки и техники / ВИНТИ, Сер. Мат. анализ. 1982.— Т. 20.— С. 179—260.— Библиогр.: с. 257—260 (57 назв.).
35. Каган Б. М., Каневский М. М. Цифровые вычислительные машины и системы.— М.: Энергия, 1973.— 679 с.
36. Калиткин Н. Н. Численные методы.— М.: Наука, 1978.— 512 с.
37. Канторович Л. В. О методе наискорейшего спуска // Докл. АН СССР.— 1947.— 56, № 3.— С. 233—236.
38. Канторович Л. В. Экономический расчет наилучшего использования ресурсов.— М.: Изд-во АН СССР, 1959.— 344 с.
39. Ким Г. Д. О статистическом исследовании ошибок округления при решении систем линейных алгебраических уравнений итерационными методами // Ошибки округления в алгебраических процессах.— М.: 1968.— С. 74—102.
40. Ким Г. Д. О распределении ошибок округления итерационных методов решения систем линейных алгебраических уравнений.— М.: Изд-во Моск. ун-та, 1969.— 114 с.
41. Ким Г. Д. О статистическом исследовании ошибок округлений некоторых алгебраических преобразований // Журн. вычисл. математики и мат. физики.— 1969.— 9, № 3.— С. 674—684.
42. Ким Г. Д. О статистическом исследовании ошибок округления при решении систем линейных уравнений итерационными методами // Вычислительные методы линейной алгебры.— Новосибирск, 1969.— С. 90—91.
43. Коллатц Л. Функциональный анализ и вычислительная математика.— М.: Мир, 1969.— 447 с.
44. Корбут Л. А., Финкельштейн Ю. Ю. Дискретное программирование.— М.: Наука, 1969.— 368 с.
45. Корнейчук Н. П. Экстремальные задачи теории приближений.— М.: Наука, 1976.— 320 с.
46. Красносельский М. А., Крейн С. Г. Итеративный процесс с минимальными невязками // Мат. сб.— 1952.— 31, № 3.— С. 315—334.
47. Красносельский М. А., Вайникко Г. М., Забрейко П. П. и др. Приближенное решение операторных уравнений.— М.: Наука, 1969.— 456 с.
48. Крылов В. И. Приближенное вычисление интегралов.— М.: Наука, 1967.— 500 с.
49. Крылов В. И., Шульгина Л. Т. Справочная книга по численному интегрированию.— М.: Наука, 1966.— 370 с.

50. Крылов В. И., Бобков В. В., Монастырский П. И. Вычислительные методы. Т. 1.— М.: Наука, 1976.— 303 с.
51. Кублановская В. Н. О применении метода Ньютона к определению собственных значений матриц // Докл. АН СССР.— 1969.— 188, № 5.— С. 1004—1005.
52. Кублановская В. Н. Применение нормализованного процесса к решению обратной проблемы собственных значений матрицы // Зап. науч. семинаров Ленингр. отд-ния мат. ин-та им. В. А. Стеклова АН СССР.— 1971.— Вып. 23.— С. 72—83.
53. Кублановская В. Н. Численные методы алгебры: Учеб. пособие.— Л.: Ленингр. кораблестроит. ин-т, 1978.— 112 с.
54. Кузнецов Ю. А. Итерационные методы решения несовместимых систем линейных уравнений // Некоторые проблемы вычислительной и прикладной математики.— Новосибирск: Наука, 1975.— С. 199—207.
55. Курош А. Г. Курс высшей алгебры: Учебник для студентов ун-тов. Изд. 11-е, стереотипное.— М.: Наука, 1975.— 431 с.
56. Ланс Дж. Н. Численные методы для быстродействующих вычислительных машин.— М.: Изд-во иностр. лит., 1962.— 208 с.
57. Ланцош К. Практические методы прикладного анализа.— М.: Физматгиз, 1961.— 524 с.
58. Лебедев В. И., Финогенов С. А. О порядке выбора итерационных параметров в чебышевском циклическом итерационном методе // Журн. вычисл. математики и мат. физики.— 1971.— 11, № 2.— С. 425—438.
59. Линник Ю. В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений.— М.: Физматгиз, 1962.— 346 с.
60. Марчук Г. И. Методы вычислительной математики.— Новосибирск: Наука, 1973.— 352 с.
61. Марчук Г. И. Методы вычислительной математики.— М.: Наука, 1977.— 455 с.
62. Михалевич В. С. Последовательные алгоритмы оптимизации и их применение // Кибернетика.— 1965.— № 1/2.— С. 85—89.
63. Моисеев Н. Н. Численные методы в теории оптимальных систем.— М.: Наука, 1971.— 424 с.
64. Молчанов И. Н., Березовская Л. И. Интерполирование и приближение функций.— Киев: —Изд-во «Знание» УССР, 1969.— 33 с.
65. Молчанов И. Н., Николенко Л. Д., Яковлев М. Ф. О решении одного класса систем линейных алгебраических уравнений с вырожденными матрицами // Вычисл. методы линейн. алгебры: Тр. Всесоюз. совещ.— Новосибирск: ВЦ СО АН СССР, 1977.— С. 97—109.
66. Молчанов И. Н. Численные методы решения некоторых задач теории упругости.— Киев: Наук. думка, 1979.— 315 с.
67. Молчанов И. Н., Яковлев М. Ф. Условия окончания итерационных процессов, гарантирующих заданную точность // Докл. АН УССР. Сер. А.— 1980.— № 6.— С. 21—23.
68. Молчанов И. Н., Тарасова Л. Г. Об одном критерии окончания итерационных процессов решения нелинейных уравнений // Там же.— 1981.— № 10.— С. 13—15.
69. Молчанов И. Н., Зубатенко В. С., Николенко Л. Д., Яковлев М. Ф. Пакет программ АРАС // Пакеты прикладных программ: Вычислительный эксперимент.— М.: Наука, 1983.— С. 129—139.
70. Морозов В. А. О задаче дифференцирования и некоторых алгоритмах приближения экспериментальной информации // Вычисл. методы и программирование.— 1970.— Вып. 14.— С. 46—62.
71. Никольский С. М. Квадратурные формулы.— М.: Наука, 1974.— 122 с.
72. Ортега Д. М., Рейнболдт В. С. Итерационные методы решения нелинейных систем уравнений со многими неизвестными.— М.: Мир, 1975.— 558 с.
73. Парлетт Б. Симметричная проблема собственных значений. Численные методы.— М.: Мир, 1983.— 382 с.
74. Полак Э. Численные методы оптимизации. Единый подход.— М.: Мир, 1974.— 376 с.
75. Поляк Б. Т. Введение в оптимизацию.— М.: Наука, 1983.— 384 с.
76. Приказчиков В. Г., Химич А. Н. Попеременно-треугольный метод вычисления главной частоты в задаче колебания пластины // Тр. V Всесоюз. конф. «Численные методы решения задач теории упругости и пластичности». Ч. 2. Новосибирск: Ин-т теорет. и прикл. механики, 1978.— С. 102—107.
77. Программное обеспечение ЭВМ МИР-1 и МИР-2. Т. 1. Численные методы.— Киев: Наук. думка, 1976.— 280 с.
78. Пшеничный Б. Н. Необходимые условия экстремума.— М.: Наука, 1969.— 151 с.
79. Пшеничный Б. Н. Метод Ньютона для решения систем равенств и неравенств // Мат. заметки.— 1970.— 8, № 5.— С. 635—640.
80. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах.— М.: Наука, 1975.— 319 с.
81. Ремез Е. Я. Основы численных методов чебышевского приближения.— Киев: Наук. думка, 1969.— 623 с.
82. Рид И., Саймон Б. Функциональный анализ. Методы современной математической физики.— М.: Мир, 1977.— 357 с.
83. Рыбаков Л. М. Метод последовательного вычисления всех действительных корней уравнения // Мат. просвещение.— 1961.— Вып. 6.— С. 262—263.
84. Савинов Г. В. Метод сопряженных градиентов для определения собственных значений // Тр. Ленингр. кораблестроит. ин-та.— 1977.— Вып. 120.— С. 55—58.
85. Самарский А. А. Введение в теорию разностных схем.— М.: Наука, 1971.— 552 с.
86. Самарский А. А. Введение в численные методы.— М.: Наука, 1982.— 271 с.
87. Самарский А. А., Николаев Е. С. Методы решения сеточных уравнений.— М.: Наука, 1978.— 559 с.
88. Стечкин С. В., Субботин Ю. Н. Сплаины в вычислительной математике.— М.: Наука, 1976.— 248 с.
89. Страховская Л. Г. Итерационный метод вычисления первой собственной функции эллиптического оператора // Журн. вычисл. математики и мат. физики.— 1977.— 17, № 3.— С. 649—664.
90. Стрелков С. П. Механика.— М.: Физматгиз, 1975.— 559 с.
91. Тихонов А. Н. Об устойчивости алгоритмов для решения вырожденных систем линейных алгебраических уравнений // Журн. вычисл. математики и мат. физики.— 1965.— 5, № 4.— С. 718—722.
92. Тихонов А. Н. О приближенных системах линейных алгебраических уравнений // Журн. вычисл. математики и мат. физики.— 1980.— 20, № 6.— С. 1373—1383.
93. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. 2-е изд. перераб. и доп.— М.: Наука, 1979.— 285 с.
94. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач.— М.: Наука, 1974.— 224 с.
95. Тьюарсон Р. Разреженные матрицы.— М.: Мир, 1977.— 190 с.
96. Уальд Д. Дж. Методы поиска экстремума.— М.: Мир, 1967.— 267 с.
97. Уилкинсон Дж. Х. Алгебраическая проблема собственных значений.— М.: Наука, 1970.— 564 с.
98. Уилкинсон Дж. Х., Райниш К. Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра.— М.: Машиностроение, 1976.— 389 с.
99. Уиттекер Э. и Робинсон Г. Математическая обработка результатов наблюдений.— М.; Л.: ОНТИ, 1935.— 360 с.
100. Фаддеева В. Н. Сдвиг для систем с плохо обусловленными матрицами // Журн. вычисл. математики и мат. физики.— 1965.— № 5.— С. 907—911.
101. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры.— Л.: Наука, 1945.— 264 с.— // Зап. науч. семинаров Ленингр. отд-ния Мат. ин-та им. В. А. Стеклова АН СССР; Т. 54/— 264 с.
102. Фаддеев Д. К., Фаддеева В. Н. О плохо обусловленных системах линейных уравнений // Журн. вычисл. математики и мат. физики.— 1961.— 1, № 3.— С. 412—417.
103. Фаддеев Д. К., Фаддеева В. Н. Вычислительные методы линейной алгебры.— М.: Физматгиз, 1963.— 736 с.
104. Фаддеев Д. К., Фаддеева В. Н. Задачи масштабирования для линейных систем // Современные численные методы: Материалы междунар. летней шк. по числ. методам, Киев, 1966.— М.; ВЦ АН СССР, 1968.— Вып. 1.— С. 76—84.

105. Фаддеев Д. К., Кублановская В. Н., Фаддеева В. Н. Линейные алгебраические системы с прямоугольными матрицами // Там же.— С. 16—75.
106. Федоренко Р. П. Приближенное решение задач оптимального управления.— М.: Наука, 1978.— 487 с.
107. Фильмаков П. Ф. Численные и графические методы прикладной математики: Справочник.— Киев: Наук. думка, 1970.— 792 с.
108. Фихтенгольц Г. М. Курс дифференциального и интегрального исчисления.— М.: Наука, 1969.— Т. 2.
109. Форсайт Дж., Моулер К. Численное решение систем линейных алгебраических уравнений.— М.: Мир, 1969.— 166 с.
110. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений.— М.: Мир, 1980.— 279 с.
111. Хемминг Р. В. Численные методы. Для научных работников и инженеров.— М.: Наука, 1972.— 400 с.
112. Химмельблауд Д. Прикладное нелинейное программирование.— М.: Мир, 1975.— 534 с.
113. Численный анализ на ФОРТРАНе: Методы и алгоритмы / Под ред. В. В. Воеводина, В. А. Морозова.— М.: Изд-во Моск. ун-та, 1980.— 84 с.
114. Численные методы условной оптимизации / Под ред. Ф. Гилла, У. Мюррея.— М.: Мир, 1977.— 290 с.
115. Шаманский В. Е. Методы численного решения краевых задач на ЭЦВМ. Линейные краевые задачи.— Киев: Наук. думка, 1963.— Ч. 1.— 196 с.
116. Шварц Л. Анализ.— М.: Мир, 1972.— Ч. 1.— 824 с.
117. Шокин Ю. И. Интервальный анализ.— Новосибирск: Наука, 1981.— 111 с.
118. Шор Н. З. Методы минимизации недифференцируемых функций и их приложения.— Киев: Наук. думка, 1979.— 200 с.
119. Эрроу К. Дж., Гурвиц Л., Удзава Х. Исследования по линейному и нелинейному программированию.— М.: Изд-во иностр. лит., 1962.— 333 с.
120. Юдин Д. В., Гольдштейн Е. Г. Линейное программирование.— М.: Наука, 1969.— 424 с.
121. Яковлев М. Ф. Замечания к явным итерационным методам решения разностных уравнений // Числен. анализ.— 1969.— Вып. 2.— С. 110—122.
122. Aasen J. O. On the reduction of a symmetric matrix to tridiagonal form // BIT (Sver).— 1971.— 11, N 3.— P. 233—242.
123. Alefeld G., Apostolatos N. Praktische Anwendung von Abschätzungsformeln bei Iterationsverfahren // Z. angew. Math. und Mech.— 1968.— 48, N 8.— S. 46—49.
124. Anselone P. M., Laurent P. J. A general method for the construction of interpolating or smoothing spline-functions // Numer. Math.— 1968.— 12, N 1.— P. 57—65.
125. Bauer F. L. Optimally scaled matrices // Ibid.— 1963.— 5, N 1.— P. 73—87.
126. Bauer F. L. Remarks on optimally scaled matrices // Ibid.— 1969.— 13, N 1.— P. 1—3.
127. Ben Israel A. An iterative method for computing the generalized inverse of an arbitrary matrix // Math. Comput.— 1965.— 19, N 91.— P. 452—455.
128. Ben Israel A. A note of an iterative method for generalized inversion of matrices // Ibid.— 1966.— 20, N 95.— P. 439—440.
129. Bohle Z. Numerical solution of inverse algebraic eigenvalue problem // Comput. J.— 1968.— 10, N 4.— P. 385—388.
130. Bradbury W. W., Fletcher R. New iterative methods for solution of the eigenproblem // Numer. Math.— 1966.— 9, N 3.— P. 259—267.
131. Broyden C. G. A class of methods for solving nonlinear simultaneous equations // Math. Comput.— 1965.— 19, N 92.— P. 577—593.
132. Broyden C. G. Quasi-Newton methods and their applications to function minimization // Ibid.— 1967.— 21, N 99.— P. 368—381.
133. Broyden C. G. A new double-rank minimization algorithm // Notic. Amer. Math. Soc.— 1969.— 16, N 4.— P. 670.
134. Broyden G. G., Dennis J. E., More J. J. On the local and superlinear convergence of quasi-Newton method // J. Inst. Math. and Appl.— 1973.— 12, N 3.— P. 223—245.
135. Bunch J. R., Kaufman L. Some stable methods for calculating inertia and solving symmetric linear systems // Math. Comput.— 1977.— 31, N 137.— P. 163—179.
136. Buzbee B. H., Golub G. H., Nielson C. W. On a direct methods for solving Poisson's equations // SIAM J. Numer. Anal.— 1970.— 7, N 4.— P. 627—656.
137. Chan S. P., Feldman R., Parlett B. N. Algorithm 517. A program for computing the conditions numbers of matrix eigenvalues without computing eigenvectors // ACM Trans. Software.— 1977.— 3, N 2.— P. 186—303.
138. Cheney E. W. Introduction to approximation theory.— New York: Acad. press, 1966.— 176 p.
139. Cheng K. Y. Minimizing the bandwidth of sparse symmetric matrices // Computing.— 1973.— 11, N 2.— P. 103—110.
140. Cline A. K., Moler C. B., Stewart C. W., Wilkinson J. H. An estimate for the condition number of a matrix // SIAM J. Numer. Anal.— 1979.— 16, N 2.— P. 368—375.
141. Davidon W. C. Variable metric method for minimization // Research and Develop. Report, A. N. L.— 5590, Argonne National Laboratory, Argonne, Illinois, 1959.— 31 p.
142. Dennis J. E., More J. J. A characterization of superlinear convergence and its application to quazi-Newton methods // Math. Comput.— 1974.— 28, N 126.— P. 549—560.
143. Dennis J. E., More J. J. Quazi-Newton methods, motivation and theory // SIAM Rev.— 1977.— 19, N 1.— P. 46—89.
144. Dennis J. E., Schnabel R. B. Numerical methods for unconstrained optimization and non-linear equations.— Englewood Cliffs (N. J.): Prentice-Hall, 1983.— 378 p.
145. Duffin R. J. The Rayleigh-Ritz method for dissipative or gyroscope systems // Quart. Appl. Math.— 1960.— 16, N 4.— P. 215—221.
146. Eispack Guide.— In: Lecture Notes in Computer Science / Smith B. T., Boyle J. M., Garbow B. S. et al. Berlin: Springer, 1974.— Vol. 6.
147. Evans D. J. The use of pre-conditioning in iterative methods for solving linear equations with symmetric positive definite matrices // J. Inst. Math. and Appl.— 1968.— 4, N 3.— P. 259—314.
148. Fletcher R. A new approach to variable metric algorithms // Comput. J.— 1970.— 13, N 3.— P. 317—322.
149. Fletcher R., Powell M. J. D. A rapidly convergent descent method for minimization // Ibid.— 1963.— 6, N 2.— P. 163—168.
150. Forsythe G. E., Wason W. R. Finite-difference methods for partial differential equations // New York; London: Wiley, 1960.— 487 p.
151. Gill P. E., Murray W. Quazi-Newton methods for unconstrained optimization // J. Inst. Math. Appl.— 1972.— 9, N 2.— P. 91—108.
152. Gill P. E., Murray W., Wright M. H. Practical Optimization.— London: Acad. press, 1981.— 536 p.
153. Goldfarb D. A family of variable-metric methods derived by variational means // Math. Comput.— 1970.— 24, N 109.— P. 23—26.
154. Golub G. H., Reinsch C. Singular values decomposition and least solution // Numer. Math.— 1970.— 14, N 5.— P. 403—420.
155. Golub G. H., Varah J. M. On a characterization of the best L_2 scaling of a matrix // SIAM J. Numer. Anal.— 1974.— 11, N 3.— P. 412—429.
156. Guderley K. G. On non-linear eigenvalue problems for matrices // SIAM J. Appl. Math.— 1958.— 6, N 4.— P. 335—353.
157. Hanson R. J. Automatic error bounds for real roots of polynomials having interval coefficients // Comput. J.— 1970.— 13, N 3.— P. 284—288.
158. Hart J. F., Cheney E. W., Lawson C. L. Computer approximation.— New York: Wiley.— 1968.— 161 p.
159. Heller D. A Survey of Parallel Algorithms in numerical linear Algebra // SIAM Rev.— 1978.— 20, N 4.— P. 741—777.
160. Hestenes M. R., Stiefel E. L. Methods of conjugate gradients for solving linear systems.— 1952.— P. 409—436. (J. Res. Nat. Bur. Stand., sect. B. Vol. 49).
161. Jennings A. Matrix computation for engineers and scientists.— New York: Wiley, 1978.— 330 p.

162. *Kahan W.* Relaxation methods for an eigenproblem.— Stanford Univ.— Stanford, 1966.— 44 p.— (Comp. Sci. Dept. / Techn. Rep. CS).
163. *Krawczyk R.* Fehlerabschätzung reeller Eigenwerte und Eigenvektoren von Matrizen // Computing.— 1969.— 4, N 4.— P. 281—283.
164. *Lanczos C.* An iterative method for the solution of the eigenvalue problem of linear differential and integral operators // J. Res. Nat. Bur. Stand., Sect. B.— 1950.— 45, N 3.— P. 255—282.
165. *Lang W.* Ein Programmsystem zur Lösung des Matrixeigenwertproblems // Wiss. Z. Techn. Hochsch. Karl-Marx-Stadt.— 1977.— 19, N. 4.— P. 437—440.
166. *Marchuk G. I., Kuznetsov Yu. A.* A stationary iterative methods for the solution of systems of linear equations with singular matrices.— Manuscript Gatlinburg 6 Symposium on Numerical Algebra.— München, 1974.— 9 p.
167. *Mayer O.* Über intervallmäßige Iterationsverfahren bei linearen Gleichungssystemen und allgemeineren Intervallgleichungssystemen // Z. angew. Math. und Mech.— 51, N 2.— P. 117—124.
168. *Meinardus G.* Approximation von Funktionen und ihre numerische Behandlung.— Berlin : Springer, 1967.— 216 S.
169. *Metropolis N. C.* Algorithms in un-normalized arithmetic: polynomial evaluation and matrix decomposition // Colloq. internat. Centre nat. rech. scient.— 1968.— N 165.— P. 293—303.
170. *Molchanov I. N.* Performance evaluation of linear algebra software // Performance Evaluation of Numerical Software (IFIP Conference).— Amsterdam : North-Holland Publ. Co. Fosdick (ed.), 1979.— P. 95—107.
171. *Molchanov I. N., Nikolenko L. D.* On an approach to integrating boundary problems with a non-unique solution // Inform. Process. Lett.— 1972.— 1, N 4.— P. 168—172.
172. *Moler C. B., Stewart G. W.* An algorithm for generalized matrix eigenvalue problems // SIAM J. Numer. Anal.— 1973.— 10, N 2.— P. 241—256.
173. *Moore E. H.* General analysis // Philadelphia Amer. Phil. Soc.— 1935.— 7, N 3.— 231 p.
174. *Osborne E. E.* On pre-conditioning of matrices // J. Assoc. Comput. Mach.— 19, 0 — 7, N 4.— P. 338—345.
175. *Ostrowski A. M.* Solution of equations and systems of equations. 2 ded.— New York ; London : Acad. Press, 1966.— 338 p.
176. *Parlett B. N., Scott D. S.* The Lanczos algorithm with selective orthogonalization.— Math. Comput.— 1979.— 33, N 145, P. 217—238.
177. *Pearson J.* On variable metric methods of minimization.— Comput. J.— 1971.— 12, N 2.— P. 171—178.
178. *Penrose R. A.* A generalized inverse for matrices // Proc. Cambridge Phil. Soc.— 1955.— 51, N 3.— P. 406—413.
179. *Peters G., Wilkinson J. H.* Eigenvalues of $Ax = \lambda Bx$ with band symmetric A and B. // Comput. J.— 1969.— 12, N 4.— P. 398—404.
180. *Peters G., Wilkinson J. H.* $Ax = \lambda Bx$ and the generalized eigenproblem // SIAM J. Numer. Anal.— 1970.— 7, N 4.— P. 479—492.
181. *Powell M. J. D.* A new algorithm for unconstrained optimization // Nonlinear Programming. / Eds Rosen J. B. et al.— New York : Acad. Press, 1970.— P. 31—65.
182. *Ralston A.* Rational Chebyshev Approximation // Mathematical methods for digital computers.— New York: Acad. Press, 1967.— Vol. 2.— P. 17—32.
183. *Rokne J.* Automatic error bounds for simple zeros of analytic functions // Comm. ACM.— 1973.— 16, N 2.— P. 101—104.
184. *Ruhe A.* Algorithms for the non-linear eigenvalue problem // SIAM J. Numer. Anal.— 1973.— 10, N 4.— P. 674—689.
185. *Ruhe A.* SOR—methods for the eigenvalue problem with large sparse matrices // Math. Comput.— 1974.— 28, N 127.— P. 696—710.
186. *Rutishauser H.* Computational aspects of F. L. Bauer's simultaneous iteration method // Numer. Math. Comput.— 1969.— 13, N 1.— P. 1—14.
187. *Schwarz H. R.* Die Reduction einer symmetrischen Bandmatrix auf tridiagonale Form // Z. angew. Math. und Mech.— 1965.— 45, N 6 (Sonder Heft).— S. 76—77.
188. *Shanno D.* Conditioning of quasi-Newton methods for function minimization // Math. Comput.— 1970.— 24, N 188.— P. 647—656.
189. *Werner H., Stoer J., Bommas W.* Rational Chebyshev approximation // Numerische Mathematik.— 1967.— Bd. 10.— P. 289—306.
190. *Wilkinson J. H.* Rounding Errors in Algebraic Processes // Notes of Applied Science, London : Her Majesty's Stationary Office, 1963.— N 32.— P. 162.
191. *Wilkinson J. H.* Global convergence of tridiagonal QR-algorithm with origin shifts // Linear Algebra and Appl.— 1968.— 1, N 3.— P. 409—420.
192. *Young D. M.* Iterative solution of large linear systems.— New York ; London : Acad. press, 1971.— 570 p.
193. *Zielke G.* Verallgemeinerte inversen, ihre anwendung und ihre effektive berechnung // Auflösungsverfahren für linearer Gleichungssysteme // Technische Hochschule Karl-Marx-Stadt.— Wissenschaftliche Informationen.— 1979, N 9.— S. 43—71.

Предисловие	3
-----------------------	---

Глава I. Прямые методы решения систем линейных алгебраических уравнений

I.1. Постановка задачи, определения, общие свойства	5
1. Задача о распределении токов в замкнутой электрической цепи (5). 2. Матрицы (7). 3. Действия над матрицами (8). 4. Определитель квадратной матрицы (9). 5. Миноры, алгебраические дополнения, ранг матрицы (10). 6. Обратная матрица (11). 7. Понятие о собственных значениях и собственных векторах (12). 8. Специальные матрицы (12). 9. Нормы векторов и матриц (15). 10. Системы линейных алгебраических уравнений (16).	
I.2. Классификация систем линейных алгебраических уравнений, описывающих прикладные задачи	18
1. Предварительные сведения (18). 2. Системы линейных алгебраических уравнений, описывающих физические модели (19). 3. Корректно и некорректно поставленные задачи (20). 4. Классификация корректно поставленных задач (22). 5. Погрешность реализации вычислительных алгоритмов на ЭВМ (24).	
I.3. Некоторые прямые методы решения систем линейных алгебраических уравнений с невырожденными матрицами	30
1. Предварительные сведения (30). 2. Метод Гаусса (31). 3. Метод, основанный на LU-разложении (компактная схема метода Гаусса) (36). 4. Метод, основанный на PU-разложении (метод отражения) (38). 5. Метод, основанный на LL^T -разложении (метод квадратных корней) (42). 6. Некоторые характеристики прямых методов решения и рекомендации по их использованию (48).	
I.4. Оценки достоверности решений, получаемых прямыми методами	50
1. Процедура итерационного уточнения решения (50). 2. Апостериорные оценки числа обусловленности системы (52).	

Глава II. Итерационные методы решения систем линейных алгебраических уравнений

II.1. Некоторые сведения из теории линейных векторных пространств	57
1. Линейное векторное пространство (57). 2. Линейная зависимость векторов (58). 3. Базис n -мерного линейного пространства (58). 4. Линейное подпространство (58). 5. Скалярное произведение векторов (59). 6. Ортогональные системы векторов (60). 7. Преобразование координат вектора при изменении базиса (61). 8. Пространство решений однородной системы (62). 9. Линейное преобразование переменных (64). 10. Собственные векторы и собственные значения матриц (66). 11. Подобные матрицы (67). 12. Билинейная и квадратичная формы матриц (68). 13. Свойства симметричных матриц (69).	
II.2. Одношаговые итерационные процессы	69
1. Предварительные замечания (69). 2. Итерационный метод Якоби (простая итерация (70). 3. Принцип построения итерационных процессов [77] (75). 4. Некоторые итерационные методы решения систем линейных алгебраических уравнений с симметричными и положительно определенными матрицами (77). 5. Ускорение сходимости итераций (83).	
II.3. Одношаговые релаксационные методы	84
1. Метод Гаусса—Зейделя (84). 2. Метод верхней релаксации (86).	
II.4. Некоторые вопросы машинной реализации одношаговых итерационных процессов	88
1. Примеры теоретически сходящихся, но не реализуемых на ЭВМ процессов (88). 2. Некоторые вопросы достоверности решений, получаемых итерационными методами (90).	

Глава III. Методы вычисления собственных значений и собственных векторов матриц

III.1. Некоторые общие сведения из линейной алгебры	98
1. Постановка задачи (98). 2. Некоторые свойства собственных значений симметричной трехдиагональной матрицы (100). 3. Ортогональные матрицы (101). 4. Элементарные матрицы вращения (102). 5. Матрицы отражения (104). 6. Каноническая форма Жордана (105). 7. Элементарные делители (107).	
III.2. Обусловленность матриц в задачах на собственные значения	107
1. Постановка вычислительных задач на собственные значения (107). 2. Обусловленность матриц при нахождении собственных значений (109). 3. Обусловленность матриц при нахождении собственных векторов (110). 4. Обусловленность при вычислении собственных значений и собственных векторов нормальных матриц (111). 5. Погрешность машинной реализации алгоритмов (111).	
III.3. Некоторые методы решения полной проблемы собственных значений	112
1. Предварительные сведения (112). 2. Метод Якоби (113). 3. Метод Хаусхолдера приведения симметричной матрицы к трехдиагональному виду (119). 4. Методы Гивенса, Шварца приведения симметричной матрицы к трехдиагональному виду (121). 5. QR- и QL-методы (124). 6. QL-метод нахождения собственных значений симметричной трехдиагональной матрицы (125). 7. Приведение матриц общего вида к форме Хессенберга (129). 8. QR-метод для вычисления собственных значений матрицы Хессенберга (131).	
III.4. Методы вычисления нескольких собственных значений и принадлежащих им собственных векторов	133
1. Метод половинного деления (133). 2. Степенной метод (136). 3. Метод скалярных произведений (139). 4. Метод обратных итераций (140). 5. Метод Ланцоша (143). 6. Метод одновременных итераций (145).	
III.5. Некоторые подходы к решению обобщенной задачи на собственные значения	147
1. Сведение к обычной задаче на собственные значения (147). 2. Приведение к обобщенной форме Шура (149).	

Глава IV. Методы решения систем линейных алгебраических уравнений с прямоугольными и квадратными вырожденными матрицами

IV.1. Некоторые предварительные сведения	154
1. Обобщенное решение для систем линейных алгебраических уравнений с прямоугольными и квадратными вырожденными матрицами (154). 2. Обращение прямоугольных и вырожденных матриц. Псевдообратная матрица (154). 3. Некоторые свойства псевдообратных матриц. Частные случаи псевдообратных матриц (157). 4. Преобразование прямоугольной матрицы A к двухдиагональному виду (158). 5. Сингулярное разложение двухдиагональной матрицы (160).	
IV.2. Решение систем с прямоугольными и квадратными вырожденными матрицами	162
1. Численное псевдообращение матриц (162). 2. Решение системы линейных алгебраических уравнений методом сингулярного разложения матрицы (163). 3. Численное решение однородных систем уравнений (165). 4. Метод регуляризации А. Н. Тихонова (166). 5. Нормализованный процесс и его применение к решению систем с произвольными прямоугольными матрицами (167). 6. Сдвиг спектра матрицы для решения систем с квадратными положительно полуопределенными матрицами (171).	
IV.3. Итерационные методы решения систем с неединственным решением и несовместных систем с симметричными и положительно полуопределенными матрицами	173
1. Методы решения систем с неединственным решением (173). 2. Итерационные методы получения обобщенного решения одного класса несовместных систем линейных алгебраических уравнений (176).	

Глава V. Методы решения нелинейных алгебраических и трансцендентных уравнений

V.1. Постановка задачи, определения, некоторые общие сведения	184
1. Задача об отражении звуковой волны на границе раздела двух сред (184). 2. Системы нелинейных уравнений (185). 3. Нелинейные уравнения с одним неизвестным (187).	
V.2. Численное решение нелинейных уравнений с одним неизвестным	189
1. Нелинейные алгебраические уравнения, описывающие физические модели (189). 2. Отделение корней (191). 3. Теоремы о неподвижной точке [82] (192). 4. Принцип построения одношаговых итерационных процессов (193). 5. Определение всех действительных корней на отрезке (198). 6. Метод секущих (199). 7. Нахождение комплексных корней трансцендентных уравнений (201). 8. Численное решение полиномиальных уравнений (204). 9. Погрешность машинной реализации (207).	
V.3. Решение систем нелинейных уравнений	209
1. Предварительные замечания (209). 2. Метод простой итерации (211). 3. Метод Ньютона (214). 4. Методы квазиьютоновского типа (215).	

Глава VI. Приближение функций

VI.1. Интерполирование и среднеквадратическое приближение функций	227
1. Основные понятия о приближении функций (227). 2. Интерполяционный полином Лагранжа (232). 3. Полином Чебышева и выбор узлов интерполирования (233). 4. Конечные разности и конечные разделенные разности (235). 5. Интерполяционная формула Ньютона (238). 6. Формула Грегори—Ньютона для интерполирования вперед (240). 7. Формула Грегори—Ньютона для интерполирования назад (243). 8. Формулы интерполирования с центральными разностями (244). 9. Экстраполирование (248). 10. Обратное интерполирование (249). 11. Интерполяционный полином Эрмита (249). 12. Интерполирование функций кубическими полиномиальными сплайнами (251). 13. Интерполирование функций двух переменных (254). 14. Среднеквадратическое приближение таблично заданных функций с помощью полиномов (255). 15. Взвешенное среднеквадратическое приближение (257). 16. Общая задача приближения по методу наименьших квадратов (258). 17. Среднеквадратическое приближение функций нескольких переменных (259).	
VI.2. Численное дифференцирование и интегрирование	260
1. Вопросы, возникающие при численном дифференцировании и интегрировании (260). 2. Численное дифференцирование с применением формулы Ньютона (263). 3. Применение формул Грегори—Ньютона (264). 4. Применение формулы Стирлинга (266). 5. Интерполяционные квадратурные формулы (267). 6. Квадратурные формулы Ньютона—Котеса (267). 7. Формулы Чебышева для численного интегрирования (270). 8. Метод квадратуры Гаусса (272).	
Список литературы	278

Монография

Игорь Николаевич Молчанов

МАШИННЫЕ МЕТОДЫ РЕШЕНИЯ ПРИКЛАДНЫХ ЗАДАЧ. АЛГЕБРА, ПРИБЛИЖЕНИЕ ФУНКЦИЙ

*Утверждено к печати ученым советом
Института кибернетики имени В. М. Глушкова АН УССР*

Редактор В. П. Егорова. Художественный редактор И. П. Антонюк.
Технический редактор А. М. Капустина. Корректоры Е. А. Михалец,
С. Д. Семенова.

ИБ № 8439

Сдано в набор 29.09.86. Подп. в печ. 12.02.87. БФ 25532. Формат 60×90/16. Бум. тип. № 1. Лит. гарн. Выс печ. Усл. печ. л. 18,0. Усл. кр.-отт. 18,0. Уч.-изд. л. 16,91. Тираж 3000 экз. Заказ 6—2925. Цена 2 р. 90 к.

Издательство «Наукова думка». 252601 Киев 4, ул. Репина, 3.

Отпечатано с матриц Головного предприятия республиканского производственного объединения «Полиграфкинг». 252057, Киев, ул. Довженко, 3 в Нестеровской городской типографии, 292310, Нестеров, Львовской обл., ул. Горького, 8. Зак. 1825